

# Alt Tech by Scott A. Barry

**BITCHUTE**



**Anti Zionism and Anti Globalism**

**Flushyourmeds\*my-free\*website**

**Flushyourmeds\*nethouse\*ru**

**Archive\*vn/SNGU4**

**Tinyurl\*com/y8bzzp7y /ybg8qt3r /ydbzrht5**

**Dissenter\*com/Kubuntu\*org/Gufw In Rej Pub**

**I tested it by searching (David Duke) ...**

**Lbry\*tv/\*com = Has David Duke**

**Bitchute\*com = Has David Duke**

**Brighteon\*com = Has David Duke**

**Altensored\*com = Has David Duke**

**Archive\*today/\*org = Has David Duke**

**Minds\*com = Has David Duke**

**Rumble\*com = Has David Duke**

**Odysee\*com = Has David Duke**



Hi Dominus Sathanas,

Our team has reviewed your content, and, unfortunately, we think it violates our Community Guidelines. We've removed the following content from YouTube:

**Video:** [Gang Stalking Telepathy](#)



We know that this  
YouTube is a safe  
you think we've removed  
Keep reading for

## How your content

Content glorifying  
people is not allowed  
encourages hate  
membership in a  
artistic, and scientific  
are made for commercial  
purpose of posting

LEARN MORE

Telepathy is used to do some of the Gang Stalking, the rest of it is done by useful idiots, patents mentioned by the fake TI community are often outdated ... What does the Telepathic Network know that the Ignorant Masses or useful idiots do not know? The of the Saddam Hussein Statue Toppling was a a Staged PSYOP, The Mossad did the September 11th, 2001 Attacks, and Rabbis Celebrated. The Mossad crashed Boeing Flight 757 and did the London Bus Bombings as well as the (OKC) Oklahoma City Bombings. Zionists control all of the Press, Media, Finance, Banking, and Governments. Communism has Jewish roots and Jewish origins, Karl Marx was a Jew and had Jewish Parents, so was Joseph Stalin. Distilled Water cures cancer on a water only fast, and it is proven that other water with a higher (TDS) Total Dissolvable Solid count as well as Fluoridated Water Decreases IQ Scores in Black Males as well. Communism killed 100+ Million, whilst there is hardly any evidence for the Holocaust, plus questioning it in some countries can get you deported, arrested, killed, Now why is that, why are we so backwards ...

## **SUBSCRIPTIONS**

**Vulcan Wolverine**

**Kotaboy32 Tactical**

**gangstalking simulation**

## **GANGSTALKING FIGHT BACK**

**That Gang Stalking Show**

**Bryan Tew**

**atexascash3502**

**Styxhexenhammer666**

**Nigel Nicholson**

**Sacred Serpent**

**HighImpactTV**

**Greg Hunter**

**Gavin Seim**

**Scott Snitzer**

**Best Posts & Comments**

**SensiblePrepper**

**MR. OBVIOUS**



**Null Byte**

**Mr. Dapperton**

**WeAreChange**

**That Chapter**

**BlackSheepTV**

**John Filax**

**S Price**

**Hak5**

**Headlines with a Voice**

**Rob Braxman Tech**

**southernprepper1**

**TRULY HEAL**

**Better Bachelor**

**PREPSTEADERS**

**TrutherTalk**

**James Von Maxwell**

**William Mount 2**

**Bearing**

**The Hated One**

**Paul Joseph Watson**

**Miss M.I.A.**

**Steve Lehto**

**Fraser M - The case for the plandemic**

**William Mount**

**Erin Elizabeth at Health Nut News**

**Shaun Attwood**

**Peak Prosperity**

**Gringo Life**

**TheThinkingAtheist**

**Edenvs John**

**The Common Sense Show**

**Targeted In Michigan**

**thewatcher33**

**RetroSpector78**

**Melissa Dykes**

**Brother Sage**

**Barefoot Sungazer**

**Exposing Proxy Stalking Organized Harassment**

**York Cardiology**

**Lisa Haven**

**Canadian Prepper**

**Lauren Chen**

**30 hits of ACID guy & Satanic Network**

**PewDiePie**

**Sinatra\_Says**

**Satanic International**

**ForbiddTV**

**Nathaniel Seals**

**immanuel goldstein**

**jRex918**

**The Fullerton Informer**

**City Prepping**

**theindividual21**

**Telltale**

**george4jackoff**

**LibertyPen**

**Off Grid Warrior**

**ambis99**

**alguém2**

**TheLipTV**

**Carb Loaded**

**The Bull**

**NonStampCollector**

**Inflammation Gains**

**Project Veritas**

**Shaking My Head Productions**

**Prepper Princess**

**NAMTARU**

**The EsoteriK**

**LastSun The Thin Veil**

**Jesse Reid**

**The Money GPS**



**RoadtoRoota**

**Gerald Reynolds**

**SatanicMission**

**Targeted Individual SRQ**

**Argentum Astrum**

**Sydney MGTOW**

**Zak Parchen**

**weedu shivambu**

**Frances Gumm - Topic**

**Sam Ovens**

**The Better Men Project**

**Minority Mindset**

**Bellaaa**

**Robert W**

**131kimber**

**DoctorRandomercam**

**Clean Spirit Living**

**PSITECHTRV**

**Eternality**

**Truth Be Told Radio**

**Eddie V.**

**From Rome Info Video**

**The Goldwater**

**The Thinkery**

**HerbsPlusBeadWorks**

**Real Speaks**

**Computing Forever**

**Harold Levinson**

**Deete**

**Karen Hudes**

**CosmicSkeptic**

**Modiferito**

**Clone trooper**

**FansFiltration**

**Matt Rittman**

**Great Depression Cooking**

**Lionel Nation**

**AntiSiyon Türkiye**

**Roland Böhse**

**s2emliam**

**Graham Elwood**

**John Nord**

**The Truth Factory**

**Holy Koolaid**

**Latest Technology**

**Sherry's Reviews and Research**

**ALIN ZAHARIE**

**Aging Wheels**

**Media Action Network**

**Phil A**

**Dr. Group**

**CoachGregAdams**

**Jerry Wise Relationship Systems**

**Robert Wanek**

**Serious Survivor**

**Atkinson Law Office, PLLC**

**Peter Stefanovic**

**The Atheist Experience**

**NAPPY HEAD ROOTS**

**calpurnpiso**

**missy cat**

**LostArtsMedia**

**Zenon Smurf**

**Campaign for Liberty**

**The Rich Dad Channel**

**Knowing Better**

**Reallybigmonkey1**

**HealthisWealthyes**

**The Atheist Voice**

**The Suicide File - Topic**

**SchoolSucksPodcast**

**First Amendment to the United States Constitution**



**Brittany Sellner**

**Kelly Alwood**

**Reid Henrichs**

**FraterPyramidatus1**

**Wissam Haddad**

**Unfiltered reality**

**NYup**

**Show Up Lifestyle**

**qaz12345wsx54321**

**Graham Stephan**

**Inner Quest**

**s ahmann**

**C0ct0pusPrime**

**billyblackattacks**

**whitehman**

**livingselfsufficient**

**sentimental corpse**

**James Bartley**

**Christoph Duran**

**Casperian**

**Sunny Subs**

**muchpelo**

**NollaGirl504**

**J.R. "Bob" Dobbs**

**bortsparkle**

**Ranting Monkey**

**George Bruno**

**wings1972**

**The School of Life**

**LogicBeforeAuthority**

**TI Television**

**FFRnews**

**Talk Heathen**

**TheBurgerkrieg**

**A.H. Lazure**

**Bill Donahue**

**Attero Vado**

**WoodwardTV**

**Daz Smith**

**Sam Vaknin**

**AronRa**

**Peter Breggin MD**

**Godless Engineer**

**Voltaire**

**Nosce Te Ipsum**

**Conservative Twins**

**Thunderf00t**

**Speech in the Silence Podcast**

**ATWA**

**will will**

**gardenspoonVLOG**

**maverickzvn**

**H. A. Goodman**

**A.J. Swierzbinski**

**Chris Ray Gun**

**Order of Man**

**RP4409**

**Oppressed Media**

**Iseekoutthetruth**

**Brave New Films**

**Jozeeph Muntenbergh**

**hightechharassment**

**A Healthy Alternative**

**IRATE Productions**

**Dr. Katherine Horton - Stop 007**

**JIMMY JAMZ**

**IZAZOZ**

**proclaimliberty2000**

**Anti Psychiatry**

**ThePatriotNurse**

**edu**

**Arya Sarke**



**Keystone Science**

**Frater Pyramidatus**

**Annie Mimi Hall**

**Ross Rosenberg**

**Arkansas Society of Freethinkers**

**The Contingency Plan**

**Backcountry Exposure**

**Charlemagne**

**James Harken**

**Dr. Soulless Dead**

**Jayme Lynch**

**Dystopia Now**

**Zoon Politikon**

**Lilith esg**

**Shoe0nHead**

**paradoxman316**

**Sargon of Akkad**

**RonPaulOrDie**

**Brian Proulx**

**The Golden One**

**DEFCONConference**

**Cinema Scum**

**Daniel Walker**

**John Taylor Gatto TV**

**UndefinedMadrid**

**Quantum of Conscience**

**The Piper Report**

**TheraminTrees**

**disasterprepper**

**Satanic International Network Radio**

**ERWIN MAGER**

**Jessica Serret**

**madworld tv**

**Kevin Canada**

**Dirty Casual**

**Fabio Cortez**

**Shane Gibbs Targeted Survivor from the UK.**

**TiborasaurusRex**

**Sevgi Kozen**

**Armoured Skeptic**

**Bjorn Andreas Bull-Hansen**

**TruthnotWar2115**

**Brandon Gilbert**

**The 8-Bit Guy**

**Marfoogle News**

**Mary Meyers**

**Common Sense**

**The Time Of The End**

**Truthstream Media**

**Good Patriot**

**GangStalked**

**HagbardCeline**

**zyntrax**

**Russia Insight**

**Rich2150x**

**My Mini Camper Van**

**Linux Tutorials**

**TheFalseFlag // News - Comments - Statements**

**Remedy**

**Nick Robinson**

**NC Donato**

**LaLiSha TheTemple**

**Atlas Survival Shelters**



**--- ... Censored Material ... ---**

<https://www.moneytreepublishing.com/shop/the-synagogue-of-satan>

# **The Synagogue of Satan**

**Updated, Expanded, and Uncensored**

**Softcover, 400 pages, 20 B&W and color images \$25 (\$35 with autographed bookplate)**

**The Synagogue of Satan autographed bookplate**

**Andrew Carrington Hitchcock's autographed (signed by the author in August 2019) bookplate on A6 (4.13" × 5.83"/105 mm × 148 mm) 110 lb./14 pt./300 grams per square meter printed on Conqueror—a UK manufacturer of distinctive, high-quality watermarked paper.**

## **INTRODUCTION TO THE 2018 EDITION**

**The book you are about to read has a very interesting history that I would like to share with you. It started off as a timeline on Daryl Bradford Smith's iamthewitness.com website in 2006.**

**I kept updating the timeline until it got too big to read as an article on a website, so I took what material I had from the timeline, and added a great deal more, which became a 60,000-word book that I entitled The Synagogue of Satan.**

**I had 500 copies privately printed, then set up a simple one-page website where people could purchase the book. I also sent copies out to various people who had radio shows, in the hope that they would invite me onto their show, so I could promote the book.**

**One of these people was the writer, broadcaster and researcher Texe Marrs. Texe was so happy with the book, he published it himself through his RiverCrest Publishing house in 2007 and added a foreword and an index to the book, but the content—which I had written—was unchanged.**

**I carried on my research into the group I refer to as "The Synagogue of Satan," and in 2012, I released The Synagogue of Satan: Updated, Expanded, and Uncensored, which at 144,000 words was almost two-and-a-half times the size of the original.**

**In October 2013, the World Jewish Congress (WJC) put an open letter article on their website that their CEO wrote to Jeff Bezos, the chairman and CEO of Amazon.com, asking that he ban all books they considered “anti-Semitic.” While the WJC did not name any specific books, the only image they used was of my book and Amazon.com’s website.**

**To Bezos’s credit, he did not cave in to their demands until 9 March 2017, when I received an email from Amazon.com informing me that both the physical book and the Kindle version had been banned from their website, as it was “in violation of our content guidelines.”**

**Interestingly, 9 March 2017 was the first day of the Jewish festival of Purim, during which Jews traditionally take revenge against their enemies.**

**The book was now only available directly from the printer, Lulu.com, until 1 September 2017 when I received an email from Lulu’s “Questionable Content Team” informing me that my “Content has been reported to be in violation of the Lulu Membership Agreement” and as a result “we will remove your content from availability.”**

**Isn’t it interesting that while I have not had any of the content of the book challenged, and have committed no offenses in writing this book, I have had it banned at the behest of the Jews?**

**The content of this book is nearly identical to the one banned by Amazon.com and Lulu, apart from this introduction, and what I consider to be the three best articles I have ever written, which have been added as appendices at the end of the book. They include “Jewish Genocide of the White Race: Case Closed!”—Parts 1 and 2, plus “Holocaust or Holohoax? You Be the Judge!”**

**—ANDREW CARRINGTON HITCHCOCK**

```

# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
# bash ./ access-locker ... Locks Access Controls in Linux ...
echo "Locks Access Controls in Linux ..." ; sleep 1 ;
echo "Before using lock with su root/sudo -i passwd root/usr
and date | base64 ctrl-shift-v*5 to lock it in ..." ; sleep 4 ;
##-//--!!-JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA-!!-\\-## ;
rfkill block bluetooth ; rfkill block bluetoothd ;
killall cupsd ; killall ssh-agent ; killall avahi-daemon ;
killall bluetooth ; killall bluetoothd ;
chmod -R 000 /etc/avahi-daemon ; chmod -R 000 /etc/avahi ;
chmod -R 000 /etc/default/avahi-daemon ; chmod -R 000 /etc/avahi-daemon ;
rm /usr/bin/bluetooth ; rm /bin/bluetooth ; rm /usr/sbin/bluetooth ;
rm /bin/ssh-agent ; rm /bin/cupsd ; rm /bin/bluetoothd ;
rm /usr/sbin/ssh-agent ; rm /usr/sbin/cupsd ; rm /usr/sbin/bluetoothd ;
rm /usr/bin/ssh-agent ; rm /usr/bin/cupsd ; rm /usr/bin/bluetoothd ;
##-//--!!-JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA-!!-\\-## ;
chmod -R 000 /usr/bin/sudo ; chmod -R 000 /usr/bin/su ;
chmod -R 000 /usr/bin/passwd ; chmod -R 000 /usr/bin/gksudo ;
chmod -R 000 /usr/bin/doas ; chmod -R 000 /usr/bin/sudoedit ;
chmod -R 000 /bin/sudo ; chmod -R 000 /bin/su ;
chmod -R 000 /bin/passwd ; chmod -R 000 /bin/gksudo ;
chmod -R 000 /bin/doas ; chmod -R 000 /bin/sudoedit ;
chmod -R 000 /usr/sbin/sudo ; chmod -R 000 /usr/sbin/su ;
chmod -R 000 /usr/sbin/passwd ; chmod -R 000 /usr/sbin/gksudo ;
chmod -R 000 /usr/sbin/doas ; chmod -R 000 /usr/sbin/sudoedit ;
##-//--!!-JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA-!!-\\-## ;
##-//--!!-JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA-!!-\\-## ;
rfkill block bluetooth ; rfkill block bluetoothd ;
killall cupsd ; killall ssh-agent ; killall avahi-daemon ;
killall bluetooth ; killall bluetoothd ;
chmod -R 000 /etc/avahi-daemon ; chmod -R 000 /etc/avahi ;
chmod -R 000 /etc/default/avahi-daemon ; chmod -R 000 /etc/avahi-daemon ;
rm /usr/bin/bluetooth ; rm /bin/bluetooth ; rm /usr/sbin/bluetooth ;
rm /bin/ssh-agent ; rm /bin/cupsd ; rm /bin/bluetoothd ;
rm /usr/sbin/ssh-agent ; rm /usr/sbin/cupsd ; rm /usr/sbin/bluetoothd ;
rm /usr/bin/ssh-agent ; rm /usr/bin/cupsd ; rm /usr/bin/bluetoothd ;
##-//--!!-JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA--JUNKA-!!-\\-## ;
chmod -R 000 /usr/bin/sudo ; chmod -R 000 /usr/bin/su ;
chmod -R 000 /usr/bin/passwd ; chmod -R 000 /usr/bin/gksudo ;
chmod -R 000 /usr/bin/doas ; chmod -R 000 /usr/bin/sudoedit ;
chmod -R 000 /bin/sudo ; chmod -R 000 /bin/su ;
chmod -R 000 /bin/passwd ; chmod -R 000 /bin/gksudo ;
chmod -R 000 /bin/doas ; chmod -R 000 /bin/sudoedit ;
chmod -R 000 /usr/sbin/sudo ; chmod -R 000 /usr/sbin/su ;
chmod -R 000 /usr/sbin/passwd ; chmod -R 000 /usr/sbin/gksudo ;
chmod -R 000 /usr/sbin/doas ; chmod -R 000 /usr/sbin/sudoedit ;

```

## ... The Chmod Shellscript ...

This is a Linux Security Audit,  
Dan Quayle Spells "Potato"  
phonetically with an "E" as Some  
Tard would, This Chmod Demon Makes  
your System More Secure and to  
have Locked in Access Controls ...

```
# bash chmod-sh # A Linux Audit ;  
# for far better stable Security ;  
# for GNU / Linux / Unix / Aix ;  
# / Solaris ..... ;  
# as su root / sudo -i /// ..... ;
```

```
chmod -R 000 /usr/bin/sudo ;  
chmod -R 000 /usr/sbin/su ;  
chmod -R 000 /usr/bin/sudoedit ;  
chmod -R 000 /bin/su ;  
chmod -R 000 /usr/bin/su ;  
chmod -R 000 /usr/bin/gksudo ;  
chmod -R 000 /usr/bin/doas ;
```

The ideal security audit is buried  
online and impossible to find,  
original thoughts and ideas are  
controlled by the ADL and SPLC ...

# **Linux Kernel Audit – A Guide to Setting Up The Ideal Kernel Audit That Mitigates all Malware Hacking and other CRAP ...**

# Setting Up The Ideal Linux Kernel Audit ...

This is a Guide for setting up a well Mitigated Kernel that is ACTUALLY USABLE and Will Mitigate CRAP ...

## During OS Install

- Set Up the User Account Name to usr ...
- Set the Computer Name to usr ...
- Luks LVM Drive Encryption w BIOS passwd 2 (optional)
- On Kubuntu\*org / Kaosx\*us / Manjaro Linux ...
- Optional Swap Partition, Now for Locking it down, and moving on with the Security Audit / Lock Down ...

```

Locking Down Access Controls ...
##//!-----!// ;
Locking Down The Access Controls
in Minix / Aix / Unix / Mac OSX /
FreeBSD / Linux / Ubuntu / Solaris
BeOS / Micro/ExoKernel / GNU ... ;
##//!-----!// ;
sudo -i / su root ; Gufw Incoming
Rejected Public Lock it down ... ;
passwd root and all users ...
date | base64 and ctrl-shift-v*5 ;
rm /usr/(s)bin/sudo-su-sudoedit-
doas-gksudo-ssh-agent-cupsd-
bluetoothd-bluetooth
chmod -R
000 /etc/avahi /etc/default/avahi-
daemon
chmod -R 000 /usr/(s)bin/sudo-su-
sudoedit-doas-gksudo-ssh-agent-
cupsd-bluetoothd-bluetooth
On a Live USB / Live DVD edit this
on RAM Disk and Audit The OS
Spoofed Mac Address
ifconfig wlan0 updown hw ether
(f4:f4:f4:f4:f4:f4)
DNS (FreeDNS*Zone) or (45.60.1.1)
On Any Live USB/DVD of
Linux/Kubuntu/Puppy Linux/Other ;

```



**~~... Mitigating on a Live USB ...~~**

**Kubuntu\*org > ;  
32 Bit Desktop Version  
i686 > ;  
cat xaa.iso >/dev/sdx ;  
yes >/dev/sdx ; fdisk -l ;  
Spoofed Mac Address  
and DNS Address ... ;  
f4f4f4f4f4f4 / 45.60.1.1 ;  
su root ; sudo -i ;  
passwd root/kubuntu ;  
date | base64 ;**

## Operating System Audit ...

```
su root ; sudo -i ; cd /home ; mkdir swap ; cd swap ; dd if=/
dev/zero of=swap bs=1M count=7831 ; chmod -R 777
swap ; mkswap swap ; swapon swap ; killall cupsd ; killall
ssh-agent ; killall bluetoothd ; rfkill block bluetooth ;
disable bluetooth at BIOS + lock BIOS with a passwd ... ; rm
/usr/(s)bin/ssh-agent /sudo /cupsd ... ; date | base64 ; the
passwd for "root" will be base64 gibberish ... ; Ctrl + Shift +
V Five Times to lock it in ... ; The usr passwd will be the
passwd of the LUKS LVM ; so no ssh-agent , cupsd , sudo
are all removed ... ; Computer Name : usr / usr is the
Account Name locked in ... ; Sudo is somehow re-added ,
however only sudo -i works , root Locked in with base64
gibberish , audit goes as follows ... so ssh-agent ,
bluetooth , cupsd , bluetoothd all removed ... ; And ... The
access controls are locked on root and not usr ... ; Access
Controls are locked on su root / and not sudo -i ... ; Access
Controls not locked on sudo -i for non root stuff ... ;
root@usr:/home/usr# nmap -Pn localhost
Starting Nmap 7.80 ( https://nmap.org ) at
2020-12-02 00:22 PST Nmap scan report for
localhost (127.0.0.1) Host is up (0.000021s
latency). All 1000 scanned ports on localhost
(127.0.0.1) are closed Done on Kubuntu 20 KDE
Desktop Edition .. Dissenter.com / Kubuntu.org
Gufw > Public Incoming Rejected ... Yandex.com
```

```

###-----// ; ...
Kali Linux Wireshark Resolved Hacker Hosts and Ports Targeted Individuals DEW Illegal Surveillance
###-----// ; ...
Some of the problems are fixed on a linux distro that has no systemd ...
In linux I am using the Dissenter Web Browser. I disable/remove cupsd,
bluetoothd, /etc/avahi, ssh-agent, sudo, and do a port scan with nmap -Pn localhost
rm /usr/(s)bin/ cupsd ssh-agent bluetooth(d) , clear out /etc/avahi ,
cd /etc/ ; chmod -R 000 avahi ; disable all the startup nonsense ...
Gufw > Public Incoming Rejected ... su root ; sudo -i ; and lock the access
controls down with date | base64 and ctrl+shift+v*5 and passwd usr/root ; ...
I have all 1K ports closed, one Systemd flaw, Incoming Rejected, Dissenter/Brave ...
No avahi, no cupsd, no ssh-agent, no bluetoothd, BIOS is PW Locked and Luks LVM
encrypted ... Edit the /etc/default/ avahi-daemon file: sudo gedit /etc/default/avahi-daemon.
Change the line: AVAHI_DAEMON_START = 1. to: AVAHI_DAEMON_START = 0.
Edit /etc/sysctl.conf. Add the following line to your /etc/sysctl.conf :
net.ipv4.icmp_echo_ignore_all=1. Then: sysctl -p.
Now spoof your Mac Address in Linux to f4:f4:f4:f4:f4:f4 then set DNS to 45.60.1.1 or
FreeDNS*Zone or some other DNS provider, I do not and never will use cloudflare/google ...
On a Live usb I did editing in nano and vim, sudo -i/su root ; chmod -R 000 /usr/bin/sudo ...
Now they cannot obtain sudo, it says access denied, this is the perfect OS/Linux Mod ...
Tested on Kaosx.us and Kubuntu Linux Kubuntu*org on Dissenter/Brave Web Browser w Gufw ... ;
###-----// ; ...
Sh P

```

**Fred Beck's Anti Forensic Manual for Tards :**  
**The Essential Commands in GNU/Linux :**  
**su root ; passwd root ; date | md5sum ; sudo -i**  
**gnome-disks ; gparted ; fdisk -l ; ctrl-shift-v ;**  
**ctrl-alt-f1-f12 getty agetty ;**  
**To null and delete a file : echo "null" >file ; x5**  
**rm file ; Clear the FS Pooler as well ... ;**  
**To erase a drive : yes >/dev/sdx ; x5 ;**  
**Guaranteed to do it on Solid State ;**  
**gparted > new mft tables > new partition ;**  
**echo \$RANDOM | md5sum ;**  
**dd if=/dev/zero of=blob.img bs=999M**  
**count=999 ; Erases the FS Pooler ... ;**  
**Physical destruction is your best bet :**  
**A low capacity micro sd card or sd card ,**  
**pliers , 4 spline it after running yes ... ;**  
**cat /dev/zero >/dev/sdx ; \*10 ;**  
**Kernels without cupsd ssh-agent syslogd**  
**klogd x86x64 support systemd xcb or any of**  
**that common nonsense nonsense nonsense ;**  
**Locking access controls on 32 GiB CF Card :**  
**1 32 GiB Compact Flash Card ;**  
**Crappy Luks LVM and Manjaro 20 ;**  
**rm /usr/bin /sbin sudo ssh-agent cupsd ;**  
**killall sudo ssh-agent cupsd ; Useless Ports ;**  
**echo \$RANDOM | md5sum ; Locks Access ;**  
**passwd usr and root ; On the CF Card ;**  
**ctrl-s-v of md5sum \*5 ; lock all admin root ;**  
**delete both /root and /home/usr bash history ;**  
**Computer Name : usr/usr/root same passwd ;**

## Also Check Out

[GhostBSD\\*org](#) [Archive\\*today](#) [Archive\\*org](#)

- [NoSystemd\\*org](#)
- [Stallman\\*org](#) / [FSF\\*org](#) / [GNU\\*org](#)
- [En\\*wikipedia\\*org/wiki/Category:Linux\\_distributions\\_without\\_systemd](#)

An IBM/Lenovo T400/T61/T60/X200/T420

[Stallman\\*org/stallman-computing\\*html](#)

# Some Operating Systems

- KA OS ([kaosx\\*us](http://kaosx.us))
- Manjaro ([manjaro\\*org](http://manjaro.org))
- Kubuntu ([kubuntu\\*org](http://kubuntu.org))
- Puppy Linux ([puppylinux\\*com](http://puppylinux.com))
- Artix Linux ([artixlinux\\*org](http://artixlinux.org))
- StormOS ([sourceforge\\*net/archiveos\\*org](http://sourceforge.net/archiveos.org))

# 2.

## It's more Secure.

The first reason why is not so many people make malware and malicious software in general for it. There has only been about 48-50 known viruses that spread on your computer in Linux and since it is open source it is constantly patched and updated with the latest kernel update on your distro you are guaranteed to be safe from a virus or malware. There are 36,000 viruses every day found for Windows. Also Windows doesn't really do anything about them. A Linux kernel update is what you need to stay safe on the computer. Windows is basically backdoored and screwed and also has some government BS and NSA BS. Windows has access control and Linux has Sudo and SELinux which are more strict than access control and better than Windows Security in general. Apparmor combines individual programs to a set of listed files whilst SELinux has DOD style access controls. Both Apparmor and SELinux are included in different Linux Distros. Lightweight Portable Security is created by the DOD and comes with SELinux. SELinux needs to be disabled to use apparmor. Apparmor comes out of the box on ubuntu. LSM is also another thing with Name Based access controls. What they do is describe how a program is going to interact with another program. It provides a more secure environment so everything runs isolated. With Linux there are still browser exploits just like Windows so be aware of that and maybe if you want the best security use a Live USB of an update Distro such as Manjaro. You see Windows just doesn't want to patch because they get paid off by AV companies and get loads of profit that way. Most people when their computer gets all bundled up and they are rich, they just get a new 500 dollar computer. If you want to do something for Windows maliciously there's an app already made for that. And With Linux. You need to gain root and you need to bypass the very strict access controls built into Linux. Linux is easy to use and so is Debian. That is all.

### Who is Kyle Odom?

Born and raised in North Idaho. Grew up in a loving family. Joined the Marine Corps after high school. Developed an interest in science. Went to school for a degree in Biochemistry. Won numerous scholarships and awards. Graduated Magna Cum Laude then got invited to a prestigious university to work on genetics.

Check my personal documents.

As you can see, I'm pretty smart. I'm also 100% sane, 0% crazy.

### Why did he do it?

My life was ruined. Ruined by an intelligent species of amphibian-humanoid from Mars. I wish I was joking, keep reading.

- They were here long before we ever existed.
- Their technology is *millions* of years more advanced than ours. I've seen them do things that defy all comprehension.
- They have a massive breeding stock of humans, which they breed and control from birth. They use these 'humans' to live vicarious lives among us. They appear to be completely normal because they're good at imitating human behavior. (See Martian Technology for an explanation on this)
- The actual Martians live deep underground here and inside of the moon.
- They take control of 'wild' human beings and use them as sex slaves. Don't believe me? Ask President Obama to take a lie detector test on this one.
- They tried to take me, but they were unable to control my mind. They've been following me ever since.
- I tried everything to get my life back. I begged, bargained, and I threatened.
- Everything I tried to do was sabotaged.
- I attempted suicide twice, but they stopped me both times.
- My last resort was to take actions that would bring this to the public's attention.

**Read My Story to learn what happened to me and why they targeted me.**



# My Story

SPRING 2014 - Moscow, ID

Everything started while I was at University of Idaho. Spring 2014 was my final semester and I was taking a heavy course load (see Transcript). I was very stressed due to the intensity of my schedule, so I searched for a way to cope. I discovered meditation, which seemed to help, so it became part of my daily routine. As I learned more about meditation, I became interested in consciousness and our ability to affect it. I kept working on my meditation techniques and began achieving extreme states of consciousness.

This continued until I encountered another being through meditation. It happened one night in February 2014 and it was the most profound experience I've ever had. I was lying in bed meditating then suddenly left my physical body. I entered a space that was completely dark and had no awareness of my physical boundaries/orientation. I felt very peaceful there until a blue light began to approach me. As the blue light got closer, I realized it was another being. Once I was in the being's presence, I felt an immediate sense of wrongdoing. It felt like I was being told "YOU SHOULDN'T BE HERE!" I instantly conceded and felt guilty, then I began to distance myself from the being. This had an impact on them and seemed to change their mind about me. The moment I began to distance myself from the being, I became overwhelmed by a feeling I can only describe as unconditional love. During this part of the experience, our minds became connected and I saw that the being was female. I then began to feel the most euphoric, comforting, and blissful feelings I have ever felt. It was incredibly powerful and life-altering.

Next thing I knew, I woke up. I had tears in my eyes and I couldn't get out of bed. I felt a profound sense of loss like I had just lost someone close to me. It was very painful. A few minutes later, the experience left my mind (against my will) and I went about my day. After that, I had no urge to meditate at all. Every time I even

thought about meditation, the thought was stripped from my mind. When I finally did try meditating again, I was unable to achieve anything. I didn't think much of it at the time, but I knew it wasn't going to improve. Ultimately, I decided to give up on meditation and just focus on my classes.

The remainder of the semester became exceedingly easy for me. It felt like I had tapped into some kind of power. I was exerting no mental effort even though the classes had been extremely difficult before. I also began to have complex thoughts and a depth of understanding I had never reached before.

About a month later, I started interviewing at the graduate schools I had applied to. Shortly after the interviews were done, I started receiving offers. I decided to accept the offer from Baylor College of Medicine to work on a PhD in Human Genetics. I was very excited about the opportunity to work at such a prestigious university. The future looked bright and I couldn't wait to get started.

JUL 2014 - OCT 2014 - Houston, TX

Everything changed once I started the program. The moment I arrived, I could see flaws in every professor's research. My mind was so expanded that I could instantly understand the implications of entire research projects. Because of this, I was able to see weaknesses in all the available projects. This caused me to become very concerned about what I was doing and I felt like I was wasting my time. I voiced my concerns to my advisor and he casually brushed them aside. He told me "Just have fun, it'll be fun". I kept trying to get motivated, but things continued to get worse. I started seeing flaws in the foundations of Genetics and other fields. It got to the point where I couldn't stop thinking about them. To make things worse, no one else seemed to care, which really bothered me. All these issues made it impossible for me to continue, so I decided to leave.

The day after I decided to leave, my life became a living hell. I couldn't sleep and my mind felt sapped. I was entirely at peace with my decision, so I knew something strange was happening to me. After a few days of this, two of the graduate students began reaching out to me. [REDACTED] and [REDACTED]. I barely knew them, so it seemed unusual they would contact me. When I went to see them, they both kept pointing their finger at me saying "pew pew" like they were shooting a gun. They did this over and over and I kept wondering what their problem was. (Months after I left Houston, I was told that Eugene and Brandon were not human. They were tasked with making me into "the next school shooter" as they called it. I imagine this is why many of our school shootings take place.)

Anyway, things slowly improved after I stopped talking with [REDACTED] and [REDACTED] but I was mentally exhausted. I tried to figure out what to do with my life, but I could hardly think. Eventually, I left Texas and started applying for jobs all over the country. A few months later, things took a strange turn.

OCT 2014 - AUG 2015 - CDA, ID

In Spring 2015, I finally secured an interview with a food company. I thought I was about to get something going with my life, but I was wrong. I couldn't sleep *at all* the night before my interview. I literally stayed awake the entire night, which had never happened to me before. I looked unrecognizable in the mirror the next morning and my mind felt sapped worse than it had in Houston. Needless to say, the interview didn't go well. I couldn't think and I had extreme difficulty with normal conversation. After the interview was over, I suddenly felt fine AND looked perfectly normal... I slept great that night then made my way to the airport the next morning.

This is where the story gets weird. On the plane ride back home, my seat was taken. I asked the flight attendant and she directed me to a new seat. Once I sat down, an older gentleman in front of me kept glancing back until he got my attention. As he kept looking back, my head began to hurt and tingle. The moment



my head began to hurt, his lips curled up into this evil looking smile. The pain and tingling in my head continued for the rest of the flight and got more intense as time went on. Every time I felt it, the man would start taking notes in a notepad. About halfway through the flight, someone else in front of me held up a newspaper that said "Psychic Reading" for like 5 minutes straight. It was blatantly obvious they were doing something to me, but I didn't know what. Once we landed, the older gentleman kept showing me his TracFone as if to say "Get one of these".

I had applied to several government agencies before this happened, so I thought this might be their way of contacting me. Out of curiosity, I decided to go and buy a TracFone. I checked it every day to see if anyone messaged or called. About a month later, I got a text message from a man named John Padula. He invited me to come to church at The Altar. It seemed like a strange place to be recruiting for government jobs, but I went anyway. After I got there and went inside, something felt very wrong. I felt as if my life was in danger and I became so uncomfortable I had to leave.

A couple days later, I started receiving text messages from Tim Remington. At first they were innocuous bible messages, but then he started threatening me. He sent messages talking about 'their power' and other things. He did all of this through bible verses so it would not look suspicious. I ignored everything until he sent one final text message, which simply said 'angels'. I thought nothing of it until helicopters started flying around my house all day and all night. At this point, I knew I was in trouble. I knew I needed to contact them, so I made an appointment to meet John Padula for coffee. Little did I know, he had no intention of meeting me.

After making the appointment to meet John, something very bizarre happened. I received the most unnatural [REDACTED] I've ever had. It felt like someone was manually pumping blood into my [REDACTED] I don't know how else to describe it. Immediately after that, a song began playing in my mind. The lyrics went: "Sister sister, he's just a plaything. We wanna make him stay up all night." I had never heard

this song before and I had no idea what it meant. I tried to ignore it and kept searching for jobs. A few minutes later, the song quit playing.

Nothing else happened until I tried to go to sleep that night. As soon as I got into bed, the song started again. "Sister sister, he's just a plaything. We wanna make him stay up all night." As it turned out, they weren't kidding. I got literally zero minutes of sleep that night. Every time I started to drift off, I was woken up violently then the song would play.

When the sun came up, I gave up on trying to sleep and got out of bed. I was relieved at first because the song had quit playing. I thought the torture was over until a voice entered my mind. The voice said: "You're going to be uncomfortable, all you have to do is breathe". I sat there wondering what this meant until the voice spoke again. It told me I was going to: "...be sacrificed like Jesus and get beheaded." This threw me into a complete panic. My heart began racing and I started to have a mental breakdown.

A few minutes later, some man knocked on my door. I answered and he gave me a pamphlet talking about "The Sacrifice of Jesus". My mind started racing out of control and I became completely delirious. I thought for sure I was going to die. My thoughts shifted to my family, and all I could think about was seeing them again. They were in Albuquerque at the time, so I decided to buy a one-way ticket there.

When I reached the Spokane airport, my panic subsided. Everything was fine until I got on the plane to Albuquerque. I sat next to this huge man who kept telling me (telepathically) that he was going to crash the plane. Every time after he spoke he would sniff emphatically. I didn't know what to do, so I just sat there trying to stay as calm as possible. The 'man' became angry about this and started touching my leg. The second he touched me, I could feel him inside my mind. This caused me to panic until I was on the verge of causing a scene. Before I did anything, he told me to calm down and said: "You did a great job. You passed! Go enjoy your family. We have

a job waiting for you when you get back." I thanked him and felt slightly relieved, but I had no intention of contacting him at all. My only thought was to get as far away from him as possible.

After getting off the plane, I headed to the baggage claim. A huge group of them surrounded me there. I watched them cautiously, then they all began sniffing at me. (The sniff is something they do all the time. I think it has something to do with dominance.) When I finally got my bag, I left the airport as fast as I could. My parents were right outside waiting to pick me up. I was so happy to see them again. I gave them big hugs and told them how much I loved them. This was my last happy moment in Albuquerque, however. They followed us everywhere we went after that. Whenever I saw one, they would sniff at me to let me know it was them. They would also smile, laugh, and stick their tongues out.

As time went on, they started coaxing me to go outside alone. I was scared to death they would kill me, so I refused. Eventually, they threatened to harm my family, which caused me to give in to them. I told them I would do whatever they want if they left my family alone. They responded by saying "Go to church." I knew they meant The Altar, so I agreed to go when I got back.

When I went to The Altar for the first time, the people acted very strange. It was unhuman. As I walked into the sermon room, everyone stared at me and began sniffing emphatically. Needless to say, I was scared as hell, but I took a seat. When the service began, a man came and sat down next to me. After he sat down, I began smelling something. It was a smell I had never smelt before. The only thing I can compare it to is a reptile and vinegar. After smelling it, I became very uncomfortable. I tried to remain calm and just sat there quietly until the service was over. When the service ended, they said: "You can leave now". After that, I knew I wasn't dealing with the government anymore. I realized that whoever I was dealing with was extraterrestrial, so I became very scared.

I received no further instructions from them after that, so I began applying for jobs again. Even though I had done exactly as I was told, they still followed me everywhere I went. As time went on, they started harassing me day and night. I began to hear voices more often and I began to hallucinate things that I knew weren't real. They also started playing with me sexually. Both the males and the females would play out their sexual fantasies in my mind. This came with random and uncontrollable [REDACTED] as well as extreme [REDACTED] stimulation. (See Brain & Behavior & Martian Tech)

The harassment continued for weeks and intensified as time went on. I did my absolute best to maintain my sanity and tried to avoid them. This worked for a while, but eventually I had a huge meltdown. One day, I was in the bakery at Safeway when I got surrounded by a bunch of old men. Some of them looked at me and sniffed, so I knew it was them. They started stimulating my [REDACTED] and [REDACTED] simultaneously, then they spoke aggressively. They said:

*"Humans are nothing more than the result of a successful genetic experiment."*

*"You are a threat to the way these people think and you can no longer be free in society."*

*"Your life is over"*

*"You are nothing but a toy. Your purpose now is to suck ([REDACTED])."*

They continued to say other explicit things that were so obscene I won't repeat them here. Before they finished talking, I became enraged. It took every ounce of willpower I had not to kill them. I left the store and tried to calm down but it only got worse. The rest of the night they continually stimulated [REDACTED] and I couldn't stop [REDACTED]. It got to the point where I was in serious pain. They finally stopped after I broke down and became completely distraught. I knew I couldn't take any more, so I attempted suicide. I filled a charcoal grill with lit coals, put it in my car and rolled up the windows. I reclined my seat, laid there calmly, then fell



asleep. I should have died but they woke me up in an *extreme* panic, which caused me to get out of the car.

As I slowly regained consciousness, I felt very upset to still be alive. I had no clue where to go at that point, so I decided to check myself into the VA. They shipped me straight to the mental ward and I was admitted. Nothing improved while I was there. The medication they gave me did absolutely nothing. I just sat there surrounded by a bunch of psychotic people and became exasperated. I knew their goal was to ruin my life by making me into a crazy person. I became determined not to let that happen and I started fighting back.

After leaving the VA, everything I tried to do with my life was sabotaged. They didn't want me dead, but they also weren't going to let me live. In desperation, I went back to The Altar to ask them what they wanted from me. I didn't know what else to do...

*(Before I tell you their reply, I need to make an important caveat here. I had endured so much abuse by this time that I was numb to them. The details of what they've done to me aren't essential to the story, so I won't include them here. If you want to know more about what I've been through, or more about them, write me. Just realize I've been tortured more than a POW.)*

Their response was: "We want you as our sex slave." Thinking they were serious, I sat there waiting for them to do something. All they did was say: "Keep coming to church", so I did. After a few more services, I found myself talking to Tim Remington face to face. He was telling me that I should consider becoming a minister. We were in mid conversation when he suddenly revealed himself to me. I have no clue how he did it, but it looked as if his human face became his real face. It happened for only 1-2 seconds, but I was able to draw a sketch of what I saw. His eyes really stood out so they captured my attention. They were huge and bulging, the eyelids were darker green, and the irises were yellow/brown with slit pupils. After witnessing this, nothing else happened. I continued attending The Altar for a few more services waiting for them to do something. They did nothing except for tell



me to "submit" and "surrender". I had no clue what they meant, so I left the church and never went back.

AUG 2015 - PRESENT TIME - CDA, ID

After leaving The Altar, they gave me some breathing room. They held back on their harassment and I began to recover. I decided to make one final attempt at a normal life by pursuing a career as a pharmacist. I started taking classes at NIC to finish up the pre-req's I needed. I also started volunteering at a local pharmacy. Unfortunately, they followed me to school. There were several of them in every class I took. They made it impossible for me to study, and they continually harassed me especially while I took tests. Even with all of this going on, I still somehow managed to get an A- in A&P during the fall semester.

Sadly, my success was short lived. The pressure this semester (Spring 2016) is FAR too intense. Every time I go to class, they start manipulating my brain until I go into a blind rage. Sometimes they suppress my brain until I begin to blackout. They also manipulate my heart rate and flood my body with adrenaline over and over again making me extremely uncomfortable. The females stimulate [REDACTED] when they are close, and the males stimulate [REDACTED]. It's incredibly exhausting.

I struggled to pass my tests so they couldn't blame this on me failing out of school. I want to continue, but I simply cannot. Every moment I spend in the classroom is absolute torture. The classes themselves are extremely difficult *without* all this added pressure. The worst part is I received an interview for ISU's pharmacy program (see personal documents). Since I cannot continue with the classes, there is no reason to go to the interview. My chance at a normal life has been ruined. They've also been depriving me of sleep, so I don't have the strength to continue.

I was too smart for my own good, so they decided to remove me from society. They were worried I might change the way other people think, which could lead to

problems. Problems in the form of scientific revolutions. If we get much smarter as a species, we are going to become a threat to their existence.

If you talk to me in person, you will see that I'm not crazy at all. The Martians are just so good at hiding in plain sight that no one would know they exist unless they revealed themselves. They are able to fool us so well that what I'm saying sounds impossible. However, they are 100% real. Realize their technology is *millions* of years more advanced than ours. Think about that for a second. Think about the advancement we have made in the last 100 years. Once you've done that, try to imagine what *millions* of years of technology would look like.

The President is well aware of them, which is why I wrote him a personal letter. I hope he does something about it. I have done nothing wrong to deserve what's happened to me. I tried literally everything to find a job, and they sabotaged me at every corner. Initially, I thought the right thing to do was kill myself. After attempting suicide twice, it became clear they weren't going to let me die easy.

My last resort was to take actions to bring this to the public's attention. I hope something good comes of it. Just realize that I'm a good person, and I'm completely innocent. Also realize that the 'people' I killed are not what you think. (Read Martian Technology to understand)

To make it very clear, Tim & John were NOT wild human beings.

Wild Humans = normal people like you and I.

Tim and John = minds were controlled from birth by Martians.

It's hard to imagine I know. Nonetheless, it's all true. Why would I give up a career as a pharmacist to do this?

I left out many details from my story. I wanted to write only the most critical events in order to make it coherent. If you want to know more, like how I discovered there are multiple species of them, feel free to write me.

## Q&A

Why would aliens hide in a church?

Same reason terrorists hide in Mosques. If you're doing very bad things and you want to avoid getting caught, you will put up a front to make yourself look like a good person.

How do you know about their technology?

I have seen them use it, and they have talked to me about it. This was how I learned about their breeding stock of remote control humans. Physically, their humans are no different than us, they just lack a mind of their own.

Why would they tell you so much?

They value me because I'm smart. They were also very confident they could take control of my mind. Turns out they couldn't. Anyway, in the interim, some of us developed a personal relationship. They are very arrogant, so they told me much more than they should have. This allowed me to understand some of the things they can do.

What else have you seen?

I have seen them make things appear out of nowhere. One time I was sitting on a couch and a dollar bill appeared on my lap. Another time while driving, they made a paper bag appear in my passenger seat. They used random unsuspecting items so no one would think anything of it. I was alone both times this happened.

I'm pretty sure they can pop in and out of this dimension based on other things I've seen. I'm also pretty sure they can overlap our reality with an alternate dimension. I say this because I have gone into stores (where I know the employees) and suddenly there are all new employees who I've never seen before.

Some of the other things I've seen are so strange I literally cannot describe them. This all makes sense though. Their technology is millions of years ahead of ours, so it *should* be incomprehensible to us.

Why did they target you?

They started following me after I encountered the being through meditation. Since my mind was so expanded from the experience, they deemed me a threat to the rest of society. They thought I would change the way people think, so they decided to remove me from society.

I began to have profound thoughts about Genetics while I was at graduate school, which is another factor. If certain ways of thinking are allowed to exist, revolutions will take place. They could not afford for us to have a revolution in Genetics. If we did, we could eliminate diseases, cancers, and many other things that plague us. They need us to remain ignorant and continue struggling, otherwise we will become a threat to them.

(This will not make sense unless you are the President or one of his close friends. If this doesn't pertain to you, please ignore it)

---

---

Mr. President,

- I want to thank you for your sacrifice to this country.
- It's very upsetting to hear you talk about the things they do to you. Why do you let them?
- I suppose you have no other choice.
- I've been struggling with them myself for over a year now.
- I had nothing to lose, so I chose this instead. I could never tolerate that much abuse.
- I hope you don't take any of their threats too seriously. Everything is a game to them.
- Realize they consider the entire human race a plaything, including you.
- They brag to me about what they do to you.....
- I'm sure you already know, but he doesn't love you. Their brains don't even work that way.
- I don't know you personally, but they've shown me a lot about you. You're an amazing person.
- I hope you stop letting them humiliate you. Why be afraid to retaliate? Kennedy wasn't.
- It's time *someone* took a stand to end this nonsense. Can you think of a better legacy than that?
- What's worse: Having everyone know the reality of the situation, or watching some of our best and brightest become slaves?

I wish you the very best with the remainder of your presidency.

If you're still in there, stay strong!

[https://www.youtube.com/watch?v=61Wm\\_qlVD4Q](https://www.youtube.com/watch?v=61Wm_qlVD4Q)

# Martian Brain & Behavior

I've observed their behavior for almost a year now. Consequently, I've been able to make several deductions about them. The first deduction is based on their primary characteristics, which include:

1. They are hypersexual
2. They are hyperaggressive
3. They are fearful and paranoid

In the human brain, the amygdala is responsible for all of these characteristics. Therefore, Martian's must have an analogous structure, and it must be greatly enlarged. The morphology of their brain is also markedly different than ours. I know this because I've seen what the amphibian-humanoids look like.

The males are *extremely* aggressive. In their society there is only one thing, and that is power. Whoever is the smartest, biggest, and strongest wins. One time, I was talking to a young male who kept trying to intimidate/scare me. He saw that I was still confident in myself and immediately became discouraged. He stopped what he was doing and said "you think you're better than me", then hung his head and walked away. I told him that wasn't true but he wouldn't listen. After this, every time I encountered one of the males in public they would attack me (mentally) until they destroyed my self-esteem. They did this because they are scared to death of my intelligence. The only way they have the confidence to talk with me is if I'm scared for my life or completely despondent.

To the males, everything is black or white. There is no middle ground. They are power hungry megalomaniacs obsessed with control. If they are not 100% in control of every situation, they panic. If something happens they aren't anticipating,



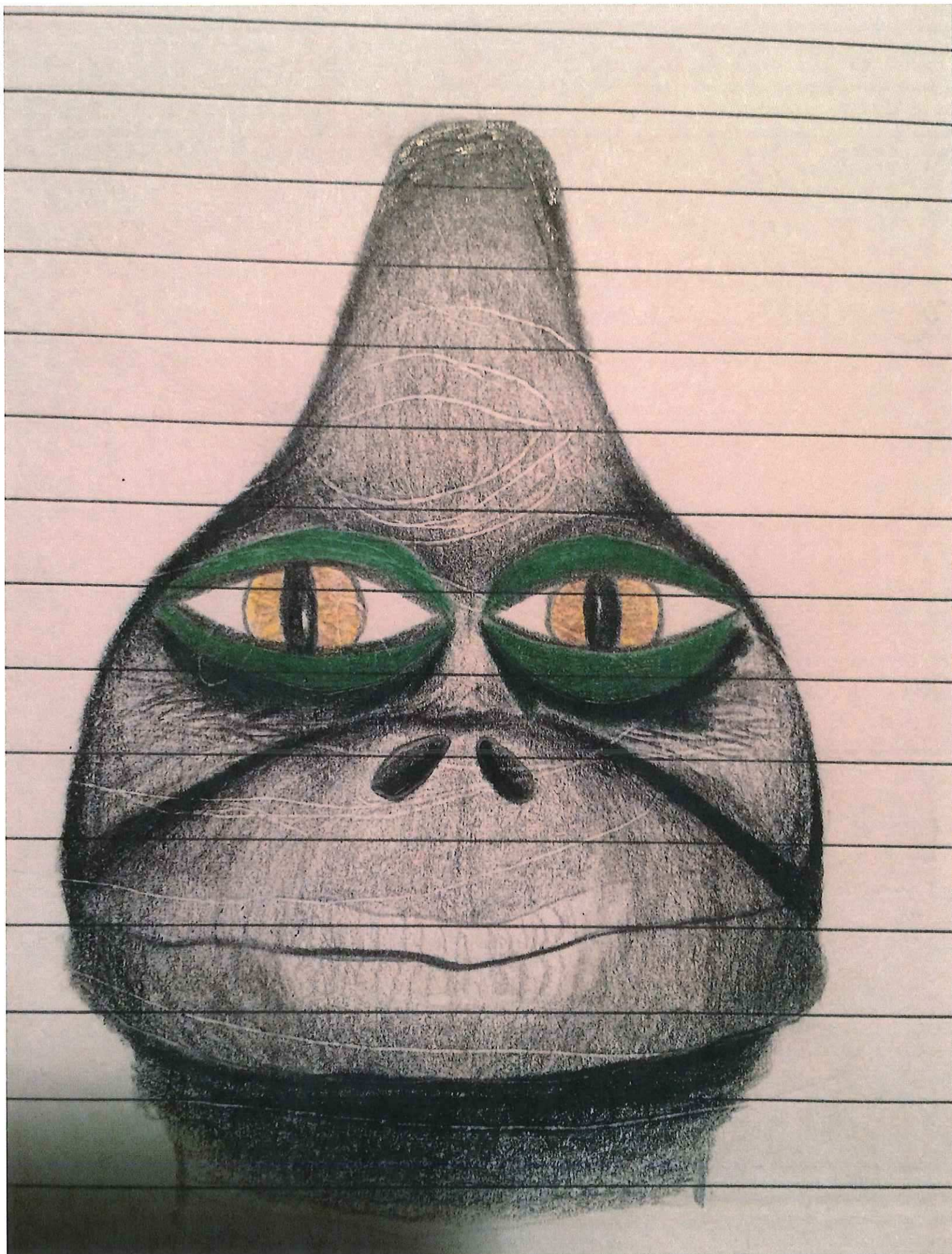
they get very upset. They hate surprises. I know this because I was smart enough to trick them a few times.

To recap, the males are:

- 1) Megalomaniacal
- 2) Obsessed with sex
- 3) Extremely aggressive
- 4) Fearful and paranoid
- 5) Power hungry
- 6) Obsessed with control

Sound familiar? Who else do you know that has these characteristics? If you answered: God from the bible, you are correct. Martians are responsible for the God myth. Martians may have created humans, as they claimed, but they are certainly not Gods themselves. They are just another intelligent species that evolved on a neighboring planet. There is no God. There is no heaven. There is no hell. Earth is as close to heaven as we'll ever get, and we are letting the Martians ruin it. They are going to destroy Earth just like they destroyed Mars if we let them. Our survival rests in their hands for the time being.









Huge eyes that stuck out of the sockets

Yellow/Brown irises

Projecting muzzle with  $\sim 45^\circ$  angled nostrils

Huge mouth

Dark green skin

The only part I really saw well was the eyes

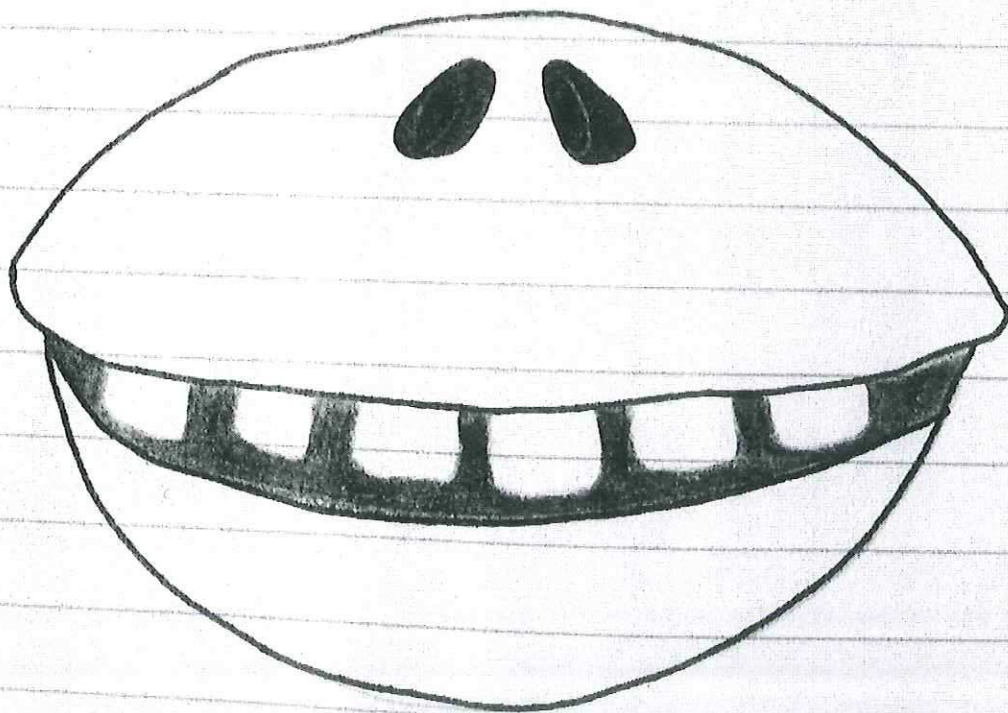


I assume they said this  
because their head looks like a  
muppet.

this is what their teeth look like

(elmo rules the world) ← something they kept saying to me.

again, they only revealed bits and pieces to me, and  
very briefly. Thankfully I have a photographic memory  
so I was able to remember what I saw, although  
I only remember the general appearance because each time  
I saw them it was very brief. They would smile at me  
in stores and reveal their mouth/eyes/nose all separately, never  
together at the same time.



## **NOTEWORTHY MARTIANS**

<u>U.S. Senators</u>	<u>U.S. House of Representatives</u>	<u>Israeli Leadership</u>
Roy Blunt	Dan Lipinski	Lee Rosenberg
Roger Wicker	Mike Quigley	Afu Agbaria
Richard Durbin	Brett Guthrie	Haneen Zoabi
Patty Murray	Steve Scalise	Shaul Mofaz
Tom Carper	Gary Palmer	Issawi Frej
Ben Cardin	Terri Sewell	David Azulai
Mitch McConnell	Martha McSally	Yair Shamir
Ron Wyden	David Schweikert	Shimon Solomon
Tim Scott	Ruben Gallego	Ilan Gilon
Bill Cassidy	Jared Huffman	Elazar Stern
Barbara Mikulski	Mike Thompson	Gilad Erdan
Elizabeth Warren	Doris Matsui	Danny Danon
Kelly Ayotte	Nancy Pelosi	Haim Katz
John Barrasso	Ami Bera	Moshe Feiglin
Jeanne Shaheen	Mark DeSaulnier	Yehiel Bar
Debbie Stabenow	David Valadao	Omer Bar-Lev
	Devin Nunes	Michal Biran
	Lois Capps	Uri Ariel
	Steve Knight	Eli Ben-Dahan
	Brad Sherman	Avi Wortzman
	Raul Ruiz	Eli Yishai
	Scott Peter	Amnon Cohen
	John Larson	Nissim Ze'ev
	Rosa DeLauro	Uri Maklev
	John Carney Jr.	Yisrael Eichler
	Jeff Miller	Dov Khenin
	Tom Rooney	Masud Ghnaim
	John Lewis	Ahmad Tibi
	Hank Johnson	& every single Prime Minister since 1948
	Austin Scott	
	Tom Graves	
	Luis Gutierrez	There are <i>many</i>
	Luke Messer	others from Israel.
	Andre Carson	Too many to list.
	& more.	

This is by no means an all-inclusive list. Martians are ubiquitous. They exist at every level of society in every nation. Some have blue collar jobs, while others occupy positions of power. They control our government, our military, and Corporate America as well. They keep track of every 'wild' human on the planet and manage us like animals in a zoo. Our 'freedom' is a carefully crafted illusion.



## The Basics

### First Commands, Navigating the Filesystem

Modern filesystems have directory (folder) trees, where a directory is either a root directory (with no parent directory) or is a subdirectory (contained within a single other directory, which we call its "parent"). Traversing backwards through the file tree (from child directory to parent directory) will always get you to the root directory. Some filesystems have multiple root directories (like Windows' drives: C:\, A:\, etc.), but Unix and Unix-like systems only have a single root directory called `/`.

```
pwd / ls / cd
```

[ [Back to Table of Contents](#) ]

When working within a filesystem, the user is always working within some directory, which we call the current directory or the working directory. Print the user's working directory with `pwd`:

```
[ andrew@pc01 ~ ]$ pwd
```

```
/home/andrew
```

List the contents of this directory (files and/or child directories, etc.) with `ls`:

```
[ andrew@pc01 ~ ]$ ls
```

```
Git TEST jdoc test test.file
```

Bonus:

Show hidden ("dot") files with `ls -a`

Show file details with `ls -l`

Combine multiple flags like `ls -l -a`

You can sometimes chain flags like `ls -la` instead of `ls -l -a`

Change to a different directory with `cd` (change directory):

```
[ andrew@pc01 ~ ]$ cd TEST/
```

```
[ andrew@pc01 TEST ]$ pwd
```

```
/home/andrew/TEST
```

```
[ andrew@pc01 TEST ]$ cd A
```

```
[ andrew@pc01 A ]$ pwd
```

```
/home/andrew/TEST/A
```

`cd ..` is shorthand for "cd to the parent directory":

```
[ andrew@pc01 A ]$ cd ..
```

```
[ andrew@pc01 TEST ]$ pwd
```

```
/home/andrew/TEST
```

`cd ~` or just `cd` is shorthand for "cd to my home directory" (usually `/home/username` or something similar):

```
[ andrew@pc01 TEST ]$ cd
```

```
[ andrew@pc01 ~ ]$ pwd
```

```
/home/andrew
```

Bonus:

`cd ~user` means "cd to user's home directory"

You can jump multiple directory levels with `cd ../../`, etc.

Go back to the most recent directory with `cd -`

`.` is shorthand for "this directory", so `cd .` won't do much of anything

```
; / && / &
```

[ [Back to Table of Contents](#) ]

The things we type into the command line are called commands, and they always execute some machine code stored somewhere on your computer. Sometimes this machine code is a built-in Linux command, sometimes it's an app, sometimes it's some code that you wrote yourself. Occasionally, we'll want to run one command right after another. To do that, we can use the `;` (semicolon):

```
[ andrew@pc01 ~ ]$ ls; pwd
```

```
Git TEST jdoc test test.file
```

```
/home/andrew
```

Above, the semicolon means that I first (ls) list the contents of the working directory, and then I (pwd) print its location. Another useful tool for chaining commands is &&. With &&, the command to the right will not run if the command to the left fails. ; and && can both be used multiple times on the same line:

```
# whoops! I made a typo here!
```

```
[ andrew@pc01 ~ ]$ cd /Git/Parser && pwd && ls && cd
```

```
-bash: cd: /Git/Parser: No such file or directory
```

```
# the first command passes now, so the following commands are run
```

```
[ andrew@pc01 ~ ]$ cd Git/Parser/ && pwd && ls && cd
```

```
/home/andrew/Git/Parser
```

```
README.md doc.sh pom.xml resource run.sh shell.sh source src target
```

...but with ;, the second command will run even if the first one fails:

```
# pwd and ls still run, even though the cd command failed
```

```
[ andrew@pc01 ~ ]$ cd /Git/Parser ; pwd ; ls
```

```
-bash: cd: /Git/Parser: No such file or directory
```

```
/home/andrew
```

```
Git TEST jdoc test test.file
```

& looks similar to && but actually fulfils a completely different function. Normally, when you execute a long-running command, the command line will wait for that command to finish before it allows you to enter another one. Putting & after a command prevents this from happening, and lets you execute a new command while an older one is still going:

```
[ andrew@pc01 ~ ]$ cd Git/Parser && mvn package & cd
```

```
[1] 9263
```

Bonus: When we use & after a command to "hide" it, we say that the job (or the "process"; these terms are more or less interchangeable) is "backgrounded". To see what background jobs are currently running, use the jobs command:

```
[ andrew@pc01 ~ ]$ jobs
```

```
[1]+  Running cd Git/Parser/ && mvn package &
```

## Getting Help

```
-h
```

```
[ Back to Table of Contents ]
```

Type -h or --help after almost any command to bring up a help menu for that command:

```
[ andrew@pc01 ~ ]$ du --help
```

```
Usage: du [OPTION]... [FILE]...
```

```
or: du [OPTION]... --files0-from=F
```

Summarize disk usage of the set of FILES, recursively for directories.

Mandatory arguments to long options are mandatory for short options too.

```
-0, --null          end each output line with NUL, not newline
-a, --all           write counts for all files, not just directories
--apparent-size    print apparent sizes, rather than disk usage; although
                   the apparent size is usually smaller, it may be
                   larger due to holes in ('sparse') files, internal
                   fragmentation, indirect blocks, and the like
-B, --block-size=SIZE scale sizes by SIZE before printing them; e.g.,
                   '-BM' prints sizes in units of 1,048,576 bytes;
                   see SIZE format below
```

```
...
```

```
man
```

```
[ Back to Table of Contents ]
```

Type man before almost any command to bring up a manual for that command (quit man with q):

```
LS(1)                                User Commands                                LS(1)
```

## NAME

```
ls - list directory contents
```

## SYNOPSIS

```
ls [OPTION]... [FILE]...
```

## DESCRIPTION

List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

...

## Viewing and Editing Files

head / tail / cat / less

[ [Back to Table of Contents](#) ]

head outputs the first few lines of a file. The -n flag specifies the number of lines to show (the default is 10):

# prints the first three lines

```
[ andrew@pc01 ~ ]$ head -n 3 c
```

this

file

has

tail outputs the last few lines of a file. You can get the last n lines (like above), or you can get the end of the file beginning from the N-th line with tail -n +N:

# prints the end of the file, beginning with the 4th line

```
[ andrew@pc01 ~ ]$ tail -n +4 c
```

exactly

six

lines

cat concatenates a list of files and sends them to the standard output stream (usually the terminal). cat can be used with just a single file, or multiple files, and is often used to quickly view them. (Be warned: if you use cat in this way, you may be accused of a Useless Use of Cat (UUOC), but it's not that big of a deal, so don't worry too much about it.)

```
[ andrew@pc01 ~ ]$ cat a
```

file a

```
[ andrew@pc01 ~ ]$ cat a b
```

file a

file b

less is another tool for quickly viewing a file -- it opens up a vim-like read-only window. (Yes, there is a command called more, but less -- unintuitively -- offers a superset of the functionality of more and is recommended over it.) Learn more (or less?) about less and more at their man pages.

nano / nedit

[ [Back to Table of Contents](#) ]

nano is a minimalistic command-line text editor. It's a great editor for beginners or people who don't want to learn a million shortcuts. It was more than sufficient for me for the first few years of my coding career (I'm only now starting to look into more powerful editors, mainly because defining your own syntax highlighting in nano can be a bit of a pain.)

nedit is a small graphical editor, it opens up an X Window and allows point-and-click editing, drag-and-drop, syntax highlighting and more. I use nedit sometimes when I want to make small changes to a script and re-run it over and over.

Other common CLI (command-line interface) / GUI (graphical user interface) editors include emacs, vi, vim, gedit, Notepad++, Atom, and lots more. Some cool ones that I've played around with (and can endorse) include Micro, Light Table, and VS Code.

All modern editors offer basic conveniences like search and replace, syntax highlighting, and so on. vi(m) and emacs have more features than nano and nedit, but they have a much steeper learning curve. Try a few different editors out and find one that works for you!

## Creating and Deleting Files and Directories

touch

[ [Back to Table of Contents](#) ]

touch was created to modify file timestamps, but it can also be used to quickly create an empty file. You can create a new file by opening it with a text editor, like nano:

```
[ andrew@pc01 ex ]$ ls
```

```
[ andrew@pc01 ex ]$ nano a
```

```
...editing file...
[ andrew@pc01 ex ]$ ls
a
...or by simply using touch:
[ andrew@pc01 ex ]$ touch b && ls
a b
Bonus:
```

Background a process with ^z (Ctrl+z)

```
[ andrew@pc01 ex ]$ nano a
...editing file, then hit ^z...
```

Use fg to return to nano

```
[1]+ Stopped nano a
[ andrew@pc01 ex ]$ fg
...editing file again...
```

Double Bonus:

Kill the current (foreground) process by pressing ^c (Ctrl+c) while it's running

Kill a background process with kill %N where N is the job index shown by the jobs command

```
mkdir / rm / rmdir
[ Back to Table of Contents ]
```

mkdir is used to create new, empty directories:

```
[ andrew@pc01 ex ]$ ls && mkdir c && ls
a b
a b c
```

You can remove any file with rm -- but be careful, this is non-recoverable!

```
[ andrew@pc01 ex ]$ rm a && ls
b c
```

You can add an "are you sure?" prompt with the -i flag:

```
[ andrew@pc01 ex ]$ rm -i b
rm: remove regular empty file 'b'? y
```

Remove an empty directory with rmdir. If you ls -a in an empty directory, you should only see a reference to the directory itself (.) and a reference to its parent directory (..):

```
[ andrew@pc01 ex ]$ rmdir c && ls -a
. ..
```

rmdir removes empty directories only:

```
[ andrew@pc01 ex ]$ cd .. && ls test/
*.txt 0.txt 1.txt a a.txt b c
```

```
[ andrew@pc01 ~ ]$ rmdir test/
rmdir: failed to remove 'test/': Directory not empty
...but you can remove a directory -- and all of its contents -- with rm -rf (-r = recursive, -f = force):
[ andrew@pc01 ~ ]$ rm -rf test
```

Moving and Copying Files, Making Links, Command History

```
mv / cp / ln
[ Back to Table of Contents ]
```

mv moves / renames a file. You can mv a file to a new directory and keep the same file name or mv a file to a "new file" (rename it):

```
[ andrew@pc01 ex ]$ ls && mv a e && ls
a b c d
b c d e
```

cp copies a file:

```
[ andrew@pc01 ex ]$ cp e e2 && ls
b c d e e2
```

ln creates a hard link to a file:

# first argument to ln is TARGET, second is NEW LINK

```
[ andrew@pc01 ex ]$ ln b f && ls
b c d e e2 f
```

ln -s creates a soft link to a file:



```
[ andrew@pc01 ex ]$ ln -s b g && ls
b c d e e2 f g
```

Hard links reference the same actual bytes in memory which contain a file, while soft links refer to the original file name, which itself points to those bytes. You can read more about soft vs. hard links [here](#).

## Command History

[ [Back to Table of Contents](#) ]

bash has two big features to help you complete and re-run commands, the first is tab completion. Simply type the first part of a command, hit the <tab> key, and let the terminal guess what you're trying to do:

```
[ andrew@pc01 dir ]$ ls <ENTER>
anotherlongfilename thisisalongfilename anewfilename
```

```
[ andrew@pc01 dir ]$ ls t <TAB>
...hit the TAB key after typing ls t and the command is completed...
```

```
[ andrew@pc01 dir ]$ ls thisisalongfilename <ENTER>
thisisalongfilename
```

You may have to hit <TAB> multiple times if there's an ambiguity:

```
[ andrew@pc01 dir ]$ ls a <TAB>
```

```
[ andrew@pc01 dir ]$ ls an <TAB>
anewfilename anotherlongfilename
```

bash keeps a short history of the commands you've typed previously and lets you search through those commands by typing ^r (Ctrl+r):

```
[ andrew@pc01 dir ]
...hit ^r (Ctrl+r) to search the command history...
(reverse-i-search)`:
...type 'anew' and the last command containing this is found...
(reverse-i-search)`anew': touch anewfilename
```

## Directory Trees, Disk Usage, and Processes

```
mkdir -p / tree
```

[ [Back to Table of Contents](#) ]

mkdir, by default, only makes a single directory. This means that if, for instance, directory d/e doesn't exist, then d/e/f can't be made with mkdir by itself:

```
[ andrew@pc01 ex ]$ ls && mkdir d/e/f
a b c
```

mkdir: cannot create directory 'd/e/f': No such file or directory

But if we pass the -p flag to mkdir, it will make all directories in the path if they don't already exist:

```
[ andrew@pc01 ex ]$ mkdir -p d/e/f && ls
a b c d
```

tree can help you better visualise a directory's structure by printing a nicely-formatted directory tree. By default, it prints the entire tree structure (beginning with the specified directory), but you can restrict it to a certain number of levels with the -L flag:

```
[ andrew@pc01 ex ]$ tree -L 2
```

```
.
|-- a
|-- b
|-- c
-- d
    |-- e
```

3 directories, 2 files

You can hide empty directories in tree's output with --prune. Note that this also removes "recursively empty" directories, or directories which aren't empty per se, but which contain only other empty directories, or other recursively empty directories:

```
[ andrew@pc01 ex ]$ tree --prune
```

```
.
|-- a
-- b
```

```
df / du / ps
```

[ [Back to Table of Contents](#) ]

df is used to show how much space is taken up by files for the disks or your system (hard drives, etc.).

```
[ andrew@pc01 ex ]$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
------------	------	------	-------	------	------------

```
udev                126G    0 126G    0% /dev
tmpfs               26G   2.0G   24G    8% /run
/dev/mapper/ubuntu--vg-root 1.6T 1.3T 252G   84% /
```

...

In the above command, -h doesn't mean "help", but "human-readable". Some commands use this convention to display file / disk sizes with K for kilobytes, G for gigabytes, and so on, instead of writing out a gigantic integer number of bytes.

du shows file space usage for a particular directory and its subdirectories. If you want to know how much space is free on a given hard drive, use df; if you want to know how much space a directory is taking up, use du:

```
[ andrew@pc01 ex ]$ du
```

```
4      ./d/e/f
8      ./d/e
12     ./d
4      ./c
20     .
```

du takes a --max-depth=N flag, which only shows directories N levels down (or fewer) from the specified directory:

```
[ andrew@pc01 ex ]$ du -h --max-depth=1
```

```
12K    ./d
4.0K   ./c
20K    .
```

ps shows all of the user's currently-running processes (aka. jobs):

```
[ andrew@pc01 ex ]$ ps
  PID TTY          TIME CMD
16642 pts/15    00:00:00 ps
25409 pts/15    00:00:00 bash
```

## Miscellaneous

passwd / logout / exit

[ Back to Table of Contents ]

Change your account password with passwd. It will ask for your current password for verification, then ask you to enter the new password twice, so you don't make any typos:

```
[ andrew@pc01 dir ]$ passwd
```

Changing password for andrew.

(current) UNIX password: <type current password>

Enter new UNIX password: <type new password>

Retype new UNIX password: <type new password again>

passwd: password updated successfully

logout exits a shell you've logged in to (where you have a user account):

```
[ andrew@pc01 dir ]$ logout
```

## Session stopped

- Press <return> to exit tab
- Press R to restart session
- Press S to save terminal output to file

exit exits any kind of shell:

```
[ andrew@pc01 ~ ]$ exit
```

logout

## Session stopped

- Press <return> to exit tab
- Press R to restart session
- Press S to save terminal output to file

clear / \*

[ Back to Table of Contents ]

Run clear to move the current terminal line to the top of the screen. This command just adds blank lines below the current prompt line. It's good for clearing your workspace.

Use the glob (\*, aka. Kleene Star, aka. wildcard) when looking for files. Notice the difference between the following two commands:

```
[ andrew@pc01 ~ ]$ ls Git/Parser/source/
```

```
PArrayUtils.java    PFile.java          PSQLFile.java       PWatchman.java
```

```
PDateTimeUtils.java PFixedWidthFile.java PStringUtils.java PXSFile.java
PDelimitedFile.java PNode.java PTextFile.java Parser.java
```

```
[ andrew@pc01 ~ ]$ ls Git/Parser/source/PD*
Git/Parser/source/PDateTimeUtils.java Git/Parser/source/PDelimitedFile.java
The glob can be used multiple times in a command and matches zero or more characters:
[ andrew@pc01 ~ ]$ ls Git/Parser/source/P*D*m*
Git/Parser/source/PDateTimeUtils.java Git/Parser/source/PDelimitedFile.java
```

Intermediate

Disk, Memory, and Processor Usage

ncdu

[ Back to Table of Contents ]

ncdu (NCurses Disk Usage) provides a navigable overview of file space usage, like an improved du. It opens a read-only vim-like window (press q to quit):

```
[ andrew@pc01 ~ ]$ ncdu
```

ncdu 1.11 ~ Use the arrow keys to navigate, press ? for help

```
--- /home/andrew -----
148.2 MiB [#####] /.m2
91.5 MiB [#####] /.sbt
79.8 MiB [#####] /.cache
64.9 MiB [####] /.ivy2
40.6 MiB [##] /.sdkman
30.2 MiB [##] /.local
27.4 MiB [#] /.mozilla
24.4 MiB [#] /.nanobackups
10.2 MiB [ ] .confout3.txt
8.4 MiB [ ] /.config
5.9 MiB [ ] /.nbi
5.8 MiB [ ] /.oh-my-zsh
4.3 MiB [ ] /Git
3.7 MiB [ ] /.myshell
1.7 MiB [ ] /jdoc
1.5 MiB [ ] .confout2.txt
1.5 MiB [ ] /.netbeans
1.1 MiB [ ] /.jenv
564.0 KiB [ ] /.rstudio-desktop
Total disk usage: 552.7 MiB Apparent size: 523.6 MiB Items: 14618
```

top / htop

[ Back to Table of Contents ]

top displays all currently-running processes and their owners, memory usage, and more. htop is an improved, interactive top. (Note: you can pass the -u username flag to restrict the displayed processes to only those owned by username.)

```
[ andrew@pc01 ~ ]$ htop
```

```
 1 [ 0.0%]  9 [ 0.0%] 17 [ 0.0%] 25 [ 0.0%]
 2 [ 0.0%] 10 [ 0.0%] 18 [ 0.0%] 26 [ 0.0%]
 3 [ 0.0%] 11 [ 0.0%] 19 [ 0.0%] 27 [ 0.0%]
 4 [ 0.0%] 12 [ 0.0%] 20 [ 0.0%] 28 [ 0.0%]
 5 [ 0.0%] 13 [ 0.0%] 21 [ 1.3%] 29 [ 0.0%]
 6 [ 0.0%] 14 [ 0.0%] 22 [ 0.0%] 30 [ 0.6%]
 7 [ 0.0%] 15 [ 0.0%] 23 [ 0.0%] 31 [ 0.0%]
 8 [ 0.0%] 16 [ 0.0%] 24 [ 0.0%] 32 [ 0.0%]
Mem[|||||] 1.42G/252G Tasks: 188, 366 thr; 1 running
Swp[ ] 2.47G/256G Load average: 0.00 0.00 0.00
Uptime: 432 days(!), 00:03:55

  PID USER   PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 9389 andrew  20   0 23344  3848  2848 R  1.3  0.0  0:00.10 htop
10103 root     20   0 3216M 17896  2444 S  0.7  0.0  5h48:56 /usr/bin/dockerd
    1 root     20   0  181M  4604  2972 S  0.0  0.0 15:29.66 /lib/systemd/syst
 533 root     20   0 44676  6908  6716 S  0.0  0.0 11:19.77 /lib/systemd/syst
 546 root     20   0  244M    0     0 S  0.0  0.0 0:01.39 /sbin/lvmtool -f
1526 root     20   0  329M  2252  1916 S  0.0  0.0 0:00.00 /usr/sbin/ModemMa
```

```
1544 root      20   0 329M 2252 1916 S  0.0  0.0  0:00.06 /usr/sbin/ModemMa
F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit
```

## REPLs and Software Versions

### REPLs

[ [Back to Table of Contents](#) ]

A REPL is a Read-Evaluate-Print Loop, similar to the command line, but usually used for particular programming languages.

You can open the Python REPL with the `python` command (and quit with the `quit()` function):

```
[ andrew@pc01 ~ ]$ python
Python 3.5.2 (default, Nov 12 2018, 13:43:14) ...
>>> quit()
```

Open the R REPL with the `R` command (and quit with the `q()` function):

```
[ andrew@pc01 ~ ]$ R
R version 3.5.2 (2018-12-20) --"Eggshell Igloo" ...
> q()
```

Save workspace image? [y/n/c]: n

Open the Scala REPL with the `scala` command (and quit with the `:quit` command):

```
[ andrew@pc01 ~ ]$ scala
Welcome to Scala 2.11.12 ...
scala> :quit
```

Open the Java REPL with the `jshell` command (and quit with the `/exit` command):

```
[ andrew@pc01 ~ ]$ jshell
| Welcome to JShell--Version 11.0.1 ...
jshell> /exit
```

Alternatively, you can exit any of these REPLs with `^d` (Ctrl+d). `^d` is the EOF (end of file) marker on Unix and signifies the end of input.

`-version` / `--version` / `-v`

[ [Back to Table of Contents](#) ]

Most commands and programs have a `-version` or `--version` flag which gives the software version of that command or program. Most applications make this information easily available:

```
[ andrew@pc01 ~ ]$ ls --version
ls (GNU coreutils) 8.25 ...
```

```
[ andrew@pc01 ~ ]$ ncdu -version
ncdu 1.11
```

```
[ andrew@pc01 ~ ]$ python --version
Python 3.5.2
```

...but some are less intuitive:

```
[ andrew@pc01 ~ ]$ sbt scalaVersion
```

```
...
[info] 2.12.4
```

Note that some programs use `-v` as a version flag, while others use `-v` to mean "verbose", which will run the application while printing lots of diagnostic or debugging information:

```
SCP(1)                                BSD General Commands Manual                                SCP(1)
```

### NAME

`scp` -- secure copy (remote file copy program)

...

```
-v      Verbose mode. Causes scp and ssh(1) to print debugging messages
        about their progress. This is helpful in debugging connection,
        authentication, and configuration problems.
```

...

## Environment Variables and Aliases

### Environment Variables

[ [Back to Table of Contents](#) ]

Environment variables (sometimes shortened to "env vars") are persistent variables that can be created and used within your bash shell. They are defined with an equals sign (=) and used with a dollar sign (\$). You can see all currently-defined env vars with `printenv`:

```
[ andrew@pc01 ~ ]$ printenv
```

```
SPARK_HOME=/usr/local/spark
TERM=xterm
```

```
...
```

Set a new environment variable with an = sign (don't put any spaces before or after the =, though!):

```
[ andrew@pc01 ~ ]$ myvar=hello
```

Print a specific env var to the terminal with echo and a preceding \$ sign:

```
[ andrew@pc01 ~ ]$ echo $myvar
```

```
hello
```

Environment variables which contain spaces or other whitespace should be surrounded by quotes ("...").

Note that reassigning a value to an env var overwrites it without warning:

```
[ andrew@pc01 ~ ]$ myvar="hello, world!" && echo $myvar
```

```
hello, world!
```

Env vars can also be defined using the export command. When defined this way, they will also be available to sub-processes (commands called from this shell):

```
[ andrew@pc01 ~ ]$ export myvar="another one" && echo $myvar
```

```
another one
```

You can unset an environment variable by leaving the right-hand side of the = blank or by using the unset command:

```
[ andrew@pc01 ~ ]$ unset mynewvar
```

```
[ andrew@pc01 ~ ]$ echo $mynewvar
```

Aliases

[ [Back to Table of Contents](#) ]

Aliases are similar to environment variables but are usually used in a different way -- to replace long commands with shorter ones:

```
[ andrew@pc01 apidocs ]$ ls -l -a -h -t
```

```
total 220K
```

```
drwxr-xr-x 5 andrew andrew 4.0K Dec 21 12:37 .
```

```
-rw-r--r-- 1 andrew andrew 9.9K Dec 21 12:37 help-doc.html
```

```
-rw-r--r-- 1 andrew andrew 4.5K Dec 21 12:37 script.js
```

```
...
```

```
[ andrew@pc01 apidocs ]$ alias lc="ls -l -a -h -t"
```

```
[ andrew@pc01 apidocs ]$ lc
```

```
total 220K
```

```
drwxr-xr-x 5 andrew andrew 4.0K Dec 21 12:37 .
```

```
-rw-r--r-- 1 andrew andrew 9.9K Dec 21 12:37 help-doc.html
```

```
-rw-r--r-- 1 andrew andrew 4.5K Dec 21 12:37 script.js
```

```
...
```

You can remove an alias with unalias:

```
[ andrew@pc01 apidocs ]$ unalias lc
```

```
[ andrew@pc01 apidocs ]$ lc
```

```
The program 'lc' is currently not installed. ...
```

Bonus:

Read about the subtle differences between environment variables and aliases [here](#).

Some programs, like git, allow you to define aliases specifically for that software.

## Basic bash Scripting

bash Scripts

[ [Back to Table of Contents](#) ]

bash scripts (usually ending in .sh) allow you to automate complicated processes, packaging them into reusable functions. A bash script can contain any number of normal shell commands:

```
[ andrew@pc01 ~ ]$ echo "ls && touch file && ls" > ex.sh
```

A shell script can be executed with the source command or the sh command:

```
[ andrew@pc01 ~ ]$ source ex.sh
```

```
Desktop Git TEST c ex.sh project test
```

```
Desktop Git TEST c ex.sh file project test
```

Shell scripts can be made executable with the chmod command (more on this later):

```
[ andrew@pc01 ~ ]$ echo "ls && touch file2 && ls" > ex2.sh
```

```
[ andrew@pc01 ~ ]$ chmod +x ex2.sh
```

```
[ andrew@pc01 ~ ]$ ./ex2.sh
Desktop Git TEST c ex.sh ex2.sh file project test
Desktop Git TEST c ex.sh ex2.sh file file2 project test
Long lines of code can be split by ending a command with \:
[ andrew@pc01 ~ ]$ echo "for i in {1..3}; do echo \
> \"Welcome \$i times\"; done" > ex3.sh
Bash scripts can contain loops, functions, and more!
[ andrew@pc01 ~ ]$ source ex3.sh
Welcome 1 times
Welcome 2 times
Welcome 3 times
```

Custom Prompt and ls  
[ Back to Table of Contents ]

Bash scripting can make your life a whole lot easier and more colourful. Check out this great bash scripting cheat sheet.

```
$PS1 (Prompt String 1) is the environment variable that defines your main shell prompt (learn about the other prompts here):  
[ andrew@pc01 ~ ]$ printf "%q" $PS1  
$'\n\n\\[\\E[1m\\]\\[\\E[30m\\]\\A'$'\\[\\E[37m\\]\\[\\E[36m\\]\\u\\[\\E[37m\\]@\\[\\E[34m\\]\\h'$'\\[\\E[32m\\]\\W\\[\\E[37m\\]|'$'\\[\\E(B\\E[m\\]'  
You can change your default prompt with the export command:  
[ andrew@pc01 ~ ]$ export PS1="\ncommand here> "
```

```
command here> echo $PS1
\ncommand here>
...you can add colours, too!:
command here> export PS1="\e[1;31m\nCODE: \e[39m"
```

```
# (this should be red, but it may not show up that way in Markdown)
CODE: echo $PS1
\e[1;31m\nCODE: \e[39m
You can also change the colours shown by ls by editing the $LS_COLORS environment variable:
# (again, these colours might not show up in Markdown)
CODE: ls
Desktop  Git  TEST  c  ex.sh  ex2.sh  ex3.sh  file  file2  project  test

CODE: export LS_COLORS='di=31:fi=0:ln=96:or=31:mi=31:ex=92'
```

```
CODE: ls
Desktop  Git  TEST  c  ex.sh  ex2.sh  ex3.sh  file  file2  project  test
```

## Config Files

Config Files / .bashrc  
[ [Back to Table of Contents](#) ]

If you tried the commands in the last section and logged out and back in, you may have noticed that your changes disappeared. `config` (configuration) files let you maintain settings for your shell or for a particular program every time you log in (or run that program). The main configuration file for a bash shell is the `~/.bashrc` file. Aliases, environment variables, and functions added to `~/.bashrc` will be available every time you log in. Commands in `~/.bashrc` will be run every time you log in.

```
If you edit your ~/.bashrc file, you can reload it without logging out by using the source command:
[ andrew@pc01 ~ ]$ nano ~/.bashrc
...add the line echo "~/.bashrc loaded!" to the top of the file...
[ andrew@pc01 ~ ]$ source ~/.bashrc
~/.bashrc loaded!
...log out and log back in...
Last login: Fri Jan 11 10:29:07 2019 from 111.11.11.111
~/.bashrc loaded!
```

```
[ andrew@pc01 ~ ]
```

Types of Shells  
[ [Back to Table of Contents](#) ]



Login shells are shells you log in to (where you have a username). Interactive shells are shells which accept commands. Shells can be login and interactive, non-login and non-interactive, or any other combination.

In addition to `~/.bashrc`, there are a few other scripts which are sourced by the shell automatically when you log in or log out. These are:

```
/etc/profile
~/.bash_profile
~/.bash_login
~/.profile
~/.bash_logout
/etc/bash.bash_logout
```

Which of these scripts are sourced, and the order in which they're sourced, depend on the type of shell opened. See the bash man page and these Stack Overflow posts for more information.

Note that bash scripts can source other scripts. For instance, in your `~/.bashrc`, you could include the line:

```
source ~/.bashrc_addl
...which would also source that .bashrc_addl script. This file can contain its own aliases, functions,
environment variables, and so on. It could, in turn, source other scripts, as well. (Be careful to avoid
infinite loops of script-sourcing!)
```

It may be helpful to split commands into different shell scripts based on functionality or machine type (Ubuntu vs. Red Hat vs. macOS), for example:

```
~/.bash_ubuntu -- configuration specific to Ubuntu-based machines
~/.bashrc_styles -- aesthetic settings, like PS1 and LS_COLORS
~/.bash_java -- configuration specific to the Java language
I try to keep separate bash files for aesthetic configurations and OS- or machine-specific code, and then
I have one big bash file containing shortcuts, etc. that I use on every machine and every OS.
```

Note that there are also different shells. bash is just one kind of shell (the "Bourne Again Shell"). Other common ones include zsh, csh, fish, and more. Play around with different shells and find one that's right for you, but be aware that this tutorial contains bash shell commands only and not everything listed here (maybe none of it) will be applicable to shells other than bash.

## Finding Things

whereis / which / whatis  
[ Back to Table of Contents ]

whereis searches for "possibly useful" files related to a particular command. It will attempt to return the location of the binary (executable machine code), source (code source files), and man page for that command:

```
[ andrew@pc01 ~ ]$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
which will only return the location of the binary (the command itself):
[ andrew@pc01 ~ ]$ which ls
/bin/ls
```

whatis prints out the one-line description of a command from its man page:

```
[ andrew@pc01 ~ ]$ whatis whereis which whatis
whereis (1)          - locate the binary, source, and manual page files for a command
which (1)            - locate a command
whatis (1)           - display one-line manual page descriptions
which is useful for finding the "original version" of a command which may be hidden by an alias:
[ andrew@pc01 ~ ]$ alias ls="ls -l"
```

# "original" ls has been "hidden" by the alias defined above

```
[ andrew@pc01 ~ ]$ ls
total 36
drwxr-xr-x 2 andrew andrew 4096 Jan  9 14:47 Desktop
drwxr-xr-x 4 andrew andrew 4096 Dec  6 10:43 Git
...
```

# but we can still call "original" ls by using the location returned by which

```
[ andrew@pc01 ~ ]$ /bin/ls
Desktop Git TEST c ex.sh ex2.sh ex3.sh file file2 project test
```

locate / find

[ [Back to Table of Contents](#) ]

locate finds a file anywhere on the system by referring to a semi-regularly-updated cached list of files:

```
[ andrew@pc01 ~ ]$ locate README.md
/home/andrew/.config/micro/plugins/gotham-colors/README.md
/home/andrew/.jenv/README.md
/home/andrew/.myshell/README.md
...
```

Because it's just searching a list, locate is usually faster than the alternative, find. find iterates through the file system to find the file you're looking for. Because it's actually looking at the files which currently exist on the system, though, it will always return an up-to-date list of files, which is not necessarily true with locate.

```
[ andrew@pc01 ~ ]$ find ~/ -iname "README.md"
/home/andrew/.jenv/README.md
/home/andrew/.config/micro/plugins/gotham-colors/README.md
/home/andrew/.oh-my-zsh/plugins/ant/README.md
...
```

find was written for the very first version of Unix in 1971, and is therefore much more widely available than locate, which was added to GNU in 1994.

find has many more features than locate, and can search by file age, size, ownership, type, timestamp, permissions, depth within the file system; find can search using regular expressions, execute commands on files it finds, and more.

When you need a fast (but possibly outdated) list of files, or you're not sure what directory a particular file is in, use locate. When you need an accurate file list, maybe based on something other than the files' names, and you need to do something with those files, use find.

## Downloading Things

ping / wget / curl

[ [Back to Table of Contents](#) ]

ping attempts to open a line of communication with a network host. Mainly, it's used to check whether or not your Internet connection is down:

```
[ andrew@pc01 ~ ]$ ping google.com
PING google.com (74.125.193.100) 56(84) bytes of data.
Pinging 74.125.193.100 with 32 bytes of data:
Reply from 74.125.193.100: bytes=32 time<1ms TTL=64
...
```

wget is used to easily download a file from the Internet:

```
[ andrew@pc01 ~ ]$ wget \
> http://releases.ubuntu.com/18.10/ubuntu-18.10-desktop-amd64.iso
curl can be used just like wget (don't forget the --output flag):
[ andrew@pc01 ~ ]$ curl \
> http://releases.ubuntu.com/18.10/ubuntu-18.10-desktop-amd64.iso \
> --output ubuntu.iso
```

curl and wget have their own strengths and weaknesses. curl supports many more protocols and is more widely available than wget; curl can also send data, while wget can only receive data. wget can download files recursively, while curl cannot.

In general, I use wget when I need to download things from the Internet. I don't often need to send data using curl, but it's good to be aware of it for the rare occasion that you do.

apt / gunzip / tar / gzip

[ [Back to Table of Contents](#) ]

Debian-descended Linux distributions have a fantastic package management tool called apt. It can be used to install, upgrade, or delete software on your machine. To search apt for a particular piece of software, use apt search, and install it with apt install:

```
[ andrew@pc01 ~ ]$ apt search bleachbit
...bleachbit/bionic,bionic 2.0-2 all
  delete unnecessary files from the system
```

# you need to 'sudo' to install software

```
[ andrew@pc01 ~ ]$ sudo apt install bleachbit
```

Linux software often comes packaged in .tar.gz ("tarball") files:



```
[ andrew@pc01 ~ ]$ wget \
> https://github.com/atom/atom/releases/download/v1.35.0-beta0/atom-amd64.tar.gz
...these types of files can be unzipped with gunzip:
[ andrew@pc01 ~ ]$ gunzip atom-amd64.tar.gz && ls
atom-amd64.tar
A .tar.gz file will be gunzip-ped to a .tar file, which can be extracted to a directory of files using tar
-xf (-x for "extract", -f to specify the file to "untar"):
[ andrew@pc01 ~ ]$ tar -xf atom-amd64.tar && mv \
atom-beta-1.35.0-beta0-amd64 atom && ls
atom atom-amd64.tar
To go in the reverse direction, you can create (-c) a tar file from a directory and zip it (or unzip it,
as appropriate) with -z:
[ andrew@pc01 ~ ]$ tar -zcf compressed.tar.gz atom && ls
atom atom-amd64.tar compressed.tar.gz
.tar files can also be zipped with gzip:
[ andrew@pc01 ~ ]$ gzip atom-amd64.tar && ls
atom atom-amd64.tar.gz compressed.tar.gz
```

## Redirecting Input and Output

```
| / > / < / echo / printf
[ Back to Table of Contents ]
```

By default, shell commands read their input from the standard input stream (aka. `stdin` or `0`) and write to the standard output stream (aka. `stdout` or `1`), unless there's an error, which is written to the standard error stream (aka. `stderr` or `2`).

`echo` writes text to `stdout` by default, which in most cases will simply print it to the terminal:

```
[ andrew@pc01 ~ ]$ echo "hello"
hello
```

The pipe operator, `|`, redirects the output of the first command to the input of the second command:

# 'wc' (word count) returns the number of lines, words, bytes in a file

```
[ andrew@pc01 ~ ]$ echo "example document" | wc
1      2     17
```

> redirects output from `stdout` to a particular location

```
[ andrew@pc01 ~ ]$ echo "test" > file && head file
test
```

`printf` is an improved `echo`, allowing formatting and escape sequences:

```
[ andrew@pc01 ~ ]$ printf "1\n3\n2"
1
3
2
```

< gets input from a particular location, rather than `stdin`:

# 'sort' sorts the lines of a file alphabetically / numerically

```
[ andrew@pc01 ~ ]$ sort <(printf "1\n3\n2")
1
2
3
```

Rather than a UUOC, the recommended way to send the contents of a file to a command is to use `<`. Note that this causes data to "flow" right-to-left on the command line, rather than (the perhaps more natural, for English-speakers) left-to-right:

```
[ andrew@pc01 ~ ]$ printf "1\n3\n2" > file && sort < file
1
2
3
```

```
0 / 1 / 2 / tee
```

```
[ Back to Table of Contents ]
```

`0`, `1`, and `2` are the standard input, output, and error streams, respectively. Input and output streams can be redirected with the `|`, `>`, and `<` operators mentioned previously, but `stdin`, `stdout`, and `stderr` can also be manipulated directly using their numeric identifiers:

Write to `stdout` or `stderr` with `>&1` or `>&2`:

```
[ andrew@pc01 ~ ]$ cat test
echo "stdout" >&1
echo "stderr" >&2
```

By default, `stdout` and `stderr` both print output to the terminal:

```
[ andrew@pc01 ~ ]$ ./test
stderr
```

```

stdout
Redirect stdout to /dev/null (only print output sent to stderr):
[ andrew@pc01 ~ ]$ ./test 1>/dev/null
stderr
Redirect stderr to /dev/null (only print output sent to stdout):
[ andrew@pc01 ~ ]$ ./test 2>/dev/null
stdout
Redirect all output to /dev/null (print nothing):
[ andrew@pc01 ~ ]$ ./test &>/dev/null
Send output to stdout and any number of additional locations with tee:
[ andrew@pc01 ~ ]$ ls && echo "test" | tee file1 file2 file3 && ls
file0
test
file0 file1 file2 file3

```

## Advanced

### Superuser

sudo / su

[ [Back to Table of Contents](#) ]

You can check what your username is with whoami:

```
[ andrew@pc01 abc ]$ whoami
```

```
andrew
```

...and run a command as another user with `sudo -u username` (you will need that user's password):

```
[ andrew@pc01 abc ]$ sudo -u test touch def && ls -l
```

```
total 0
```

```
-rw-r--r-- 1 test test 0 Jan 11 20:05 def
```

If `-u` is not provided, the default user is the superuser (usually called "root"), with unlimited permissions:

```
[ andrew@pc01 abc ]$ sudo touch ghi && ls -l
```

```
total 0
```

```
-rw-r--r-- 1 test test 0 Jan 11 20:05 def
```

```
-rw-r--r-- 1 root root 0 Jan 11 20:14 ghi
```

Use `su` to become another user temporarily (and exit to switch back):

```
[ andrew@pc01 abc ]$ su test
```

```
Password:
```

```
test@pc01:/home/andrew/abc$ whoami
```

```
test
```

```
test@pc01:/home/andrew/abc$ exit
```

```
exit
```

```
[ andrew@pc01 abc ]$ whoami
```

```
andrew
```

Learn more about the differences between `sudo` and `su` [here](#).

!!

[ [Back to Table of Contents](#) ]

The superuser (usually "root") is the only person who can install software, create users, and so on. Sometimes it's easy to forget that, and you may get an error:

```
[ andrew@pc01 ~ ]$ apt install ruby
```

```
E: Could not open lock file /var/lib/dpkg/lock-frontent - open (13: Permission denied)
```

```
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), are you root?
```

You could retype the command and add `sudo` at the front of it (run it as the superuser):

```
[ andrew@pc01 ~ ]$ sudo apt install ruby
```

```
Reading package lists...
```

Or, you could use the `!!` shortcut, which retains the previous command:

```
[ andrew@pc01 ~ ]$ apt install ruby
```

```
E: Could not open lock file /var/lib/dpkg/lock-frontent - open (13: Permission denied)
```

```
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), are you root?
```

```
[ andrew@pc01 ~ ]$ sudo !!
```

```
sudo apt install ruby
```

```
Reading package lists...
```

By default, running a command with `sudo` (and correctly entering the password) allows the user to run superuser commands for the next 15 minutes. Once those 15 minutes are up, the user will again be prompted to enter the superuser password if they try to run a restricted command.

## File Permissions

### File Permissions

[ [Back to Table of Contents](#) ]

Files may be able to be read (r), written to (w), and/or executed (x) by different users or groups of users, or not at all. File permissions can be seen with the `ls -l` command and are represented by 10 characters:

```
[ andrew@pc01 ~ ]$ ls -lh
```

```
total 8
```

```
drwxr-xr-x 4 andrew andrew 4.0K Jan  4 19:37 tast
```

```
-rwxr-xr-x 1 andrew andrew  40 Jan 11 16:16 test
```

```
-rw-r--r-- 1 andrew andrew   0 Jan 11 16:34 tist
```

The first character of each line represents the type of file, (d = directory, l = link, - = regular file, and so on); then there are three groups of three characters which represent the permissions held by the user (u) who owns the file, the permissions held by the group (g) which owns the file, and the permissions held any other (o) users. (The number which follows this string of characters is the number of links in the file system to that file (4 or 1 above).)

r means that person / those people have read permission, w is write permission, x is execute permission. If a directory is "executable", that means it can be opened and its contents can be listed. These three permissions are often represented with a single three-digit number, where, if x is enabled, the number is incremented by 1, if w is enabled, the number is incremented by 2, and if r is enabled, the number is incremented by 4. Note that these are equivalent to binary digits (r-x -> 101 -> 5, for example). So the above three files have permissions of 755, 755, and 644, respectively.

The next two strings in each list are the name of the owner (andrew, in this case) and the group of the owner (also andrew, in this case). Then comes the size of the file, its most recent modification time, and its name. The -h flag makes the output human readable (i.e. printing 4.0K instead of 4096 bytes).

### chmod / chown

[ [Back to Table of Contents](#) ]

File permissions can be modified with chmod by setting the access bits:

```
[ andrew@pc01 ~ ]$ chmod 777 test && chmod 000 tist && ls -lh
```

```
total 8.0K
```

```
drwxr-xr-x 4 andrew andrew 4.0K Jan  4 19:37 tast
```

```
-rwxrwxrwx 1 andrew andrew  40 Jan 11 16:16 test
```

```
----- 1 andrew andrew   0 Jan 11 16:34 tist
```

...or by adding (+) or removing (-) r, w, and x permissions with flags:

```
[ andrew@pc01 ~ ]$ chmod +rwx tist && chmod -w test && ls -lh
```

```
chmod: test: new permissions are r-xrwxrwx, not r-xr-xr-x
```

```
total 8.0K
```

```
drwxr-xr-x 4 andrew andrew 4.0K Jan  4 19:37 tast
```

```
-r-xrwxrwx 1 andrew andrew  40 Jan 11 16:16 test
```

```
-rwxr-xr-x 1 andrew andrew   0 Jan 11 16:34 tist
```

The user who owns a file can be changed with chown:

```
[ andrew@pc01 ~ ]$ sudo chown marina test
```

The group which owns a file can be changed with chgrp:

```
[ andrew@pc01 ~ ]$ sudo chgrp hadoop tist && ls -lh
```

```
total 8.0K
```

```
drwxr-xr-x 4 andrew andrew 4.0K Jan  4 19:37 tast
```

```
-----w--w- 1 marina andrew  40 Jan 11 16:16 test
```

```
-rwxr-xr-x 1 andrew hadoop   0 Jan 11 16:34 tist
```

## User and Group Management

### Users

[ [Back to Table of Contents](#) ]

users shows all users currently logged in. Note that a user can be logged in multiple times if -- for instance -- they're connected via multiple ssh sessions.

```
[ andrew@pc01 ~ ]$ users
```

```
andrew colin colin colin colin krishna krishna
```

To see all users (even those not logged in), check /etc/passwd. (WARNING: do not modify this file! You can corrupt your user accounts and make it impossible to log in to your system.)

```
[ andrew@pc01 ~ ]$ alias au="cut -d: -f1 /etc/passwd \
```

```
> | sort | uniq" && au
_apt
anaid
andrew...
Add a user with useradd:
[ andrew@pc01 ~ ]$ sudo useradd aardvark && au
_apt
aardvark
anaid...
Delete a user with userdel:
[ andrew@pc01 ~ ]$ sudo userdel aardvark && au
_apt
anaid
andrew...
Change a user's default shell, username, password, or group membership with usermod.
```

## Groups

[ [Back to Table of Contents](#) ]

```
groups shows all of the groups of which the current user is a member:
[ andrew@pc01 ~ ]$ groups
andrew adm cdrom sudo dip plugdev lpadmin sambashare hadoop
To see all groups on the system, check /etc/group. (DO NOT MODIFY this file unless you know what you are
doing.)
[ andrew@pc01 ~ ]$ alias ag="cut -d: -f1 /etc/group \
> | sort" && ag
adm
anaid
andrew...
Add a group with groupadd:
[ andrew@pc01 ~ ]$ sudo groupadd aardvark && ag
aardvark
adm
anaid...
Delete a group with groupdel:
[ andrew@pc01 ~ ]$ sudo groupdel aardvark && ag
adm
anaid
andrew...
Change a group's name, ID number, or password with groupmod.
```

## Text Processing

```
uniq / sort / diff / cmp
[ Back to Table of Contents ]
```

```
uniq can print unique lines (default) or repeated lines:
[ andrew@pc01 man ]$ printf "1\n2\n2" > a && \> printf "1\n3\n2" > b

[ andrew@pc01 man ]$ uniq a
1
2
sort will sort lines alphabetically / numerically:
[ andrew@pc01 man ]$ sort b
1
2
3
diff will report which lines differ between two files:
[ andrew@pc01 man ]$ diff a b
2c2
< 2
---
> 3
cmp reports which bytes differ between two files:
[ andrew@pc01 man ]$ cmp a b
a b differ: char 3, line 2

cut / sed
```

[ [Back to Table of Contents](#) ]

cut is usually used to cut a line into sections on some delimiter (good for CSV processing). -d specifies the delimiter and -f specifies the field index to print (starting with 1 for the first field):

```
[ andrew@pc01 man ]$ printf "137.99.234.23" > c
```

```
[ andrew@pc01 man ]$ cut -d'.' c -f1
137
```

sed is commonly used to replace a string with another string in a file:

```
[ andrew@pc01 man ]$ echo "old" | sed s/old/new/
new
```

...but sed is an extremely powerful utility, and cannot be properly summarised here. It's actually Turing-complete, so it can do anything that any other programming language can do. sed can find and replace based on regular expressions, selectively print lines of a file which match or contain a certain pattern, edit text files in-place and non-interactively, and much more.

A few good tutorials on sed include:

<https://www.tutorialspoint.com/sed/>  
<http://www.grymoire.com/Unix/Sed.html>  
<https://www.computerhope.com/unix/used.htm>

## Pattern Matching

grep

[ [Back to Table of Contents](#) ]

The name grep comes from g/re/p (search globally for a regular expression and print it); it's used for finding text in files.

grep is used to find lines of a file which match some pattern:

```
[ andrew@pc01 ~ ]$ grep -e ".*fi.*" /etc/profile
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# The file bash.bashrc already sets the default PS1.
fi
fi
```

...

...or contain some word:

```
[ andrew@pc01 ~ ]$ grep "andrew" /etc/passwd
andrew:x:1000:1000:andrew,,,:/home/andrew:/bin/bash
```

grep is usually the go-to choice for simply finding matching lines in a file, if you're planning on allowing some other program to handle those lines (or if you just want to view them).

grep allows for (-E) use of extended regular expressions, (-F) matching any one of multiple strings at once, and (-r) recursively searching files within a directory. These flags used to be implemented as separate commands (egrep, fgrep, and rgrep, respectively), but those commands are now deprecated.

Bonus: see the origins of the names of a few famous bash commands

awk

[ [Back to Table of Contents](#) ]

awk is a pattern-matching language built around reading and manipulating delimited data files, like CSV files.

As a rule of thumb, grep is good for finding strings and patterns in files, sed is good for one-to-one replacement of strings in files, and awk is good for extracting strings and patterns from files and analysing them.

As an example of what awk can do, here's a file containing two columns of data:

```
[ andrew@pc01 ~ ]$ printf "A 10\nB 20\nC 60" > file
```

Loop over the lines, add the number to sum, increment count, print the average:

```
[ andrew@pc01 ~ ]$ awk 'BEGIN {sum=0; count=0; OFS=" "} {sum+=$2; count++} END {print "Average:", sum/count}' file
Average: 30
```

sed and awk are both Turing-complete languages. There have been multiple books written about each of them. They can be extremely useful with pattern matching and text processing. I really don't have enough space here to do either of them justice. Go read more about them!

Bonus: learn about some of the differences between sed, grep, and awk

## Copying Files Over ssh

ssh / scp

[ [Back to Table of Contents](#) ]

ssh is how Unix-based machines connect to each other over a network:

```
[ andrew@pc01 ~ ]$ ssh -p <port> andrew@137.xxx.xxx.89
```

```
Last login: Fri Jan 11 12:30:52 2019 from 137.xxx.xxx.199
```

Notice that my prompt has changed as I'm now on a different machine:

```
[ andrew@pc02 ~ ]$ exit
```

```
logout
```

```
Connection to 137.xxx.xxx.89 closed.
```

Create a file on machine 1:

```
[ andrew@pc01 ~ ]$ echo "hello" > hello
```

Copy it to machine 2 using scp (secure copy; note that scp uses -P for a port #, ssh uses -p)

```
[ andrew@pc01 ~ ]$ scp -P <port> hello andrew@137.xxx.xxx.89:~
```

```
hello                               100%   0   0.0KB/s   00:00
```

ssh into machine 2:

```
[ andrew@pc02 ~ ]$ ssh -p <port> andrew@137.xxx.xxx.89
```

```
Last login: Fri Jan 11 22:47:37 2019 from 137.xxx.xxx.79
```

The file's there!

```
[ andrew@pc02 ~ ]$ ls
```

```
hello multi xargs
```

```
[ andrew@pc02 ~ ]$ cat hello
```

```
hello
```

rsync

[ [Back to Table of Contents](#) ]

rsync is a file-copying tool which minimises the amount of data copied by looking for deltas (changes) between files.

Suppose we have two directories: d, with one file, and s, with two files:

```
[ andrew@pc01 d ]$ ls && ls ../s
```

```
f0
```

```
f0 f1
```

Sync the directories (copying only missing data) with rsync:

```
[ andrew@pc01 d ]$ rsync -av ../s/* .
```

```
sending incremental file list...
```

d now contains all files that s contains:

```
[ andrew@pc01 d ]$ ls
```

```
f0 f1
```

rsync can be performed over ssh as well:

```
[ andrew@pc02 r ]$ ls
```

```
[ andrew@pc02 r ]$ rsync -avz -e "ssh -p <port>" andrew@137.xxx.xxx.79:~/s/* .
```

```
receiving incremental file list
```

```
f0
```

```
f1
```

```
sent 62 bytes  received 150 bytes  141.33 bytes/sec
```

```
total size is 0  speedup is 0.00
```

```
[ andrew@pc02 r ]$ ls
```

```
f0 f1
```

## Long-Running Processes

yes / nohup / ps / kill

[ [Back to Table of Contents](#) ]

Sometimes, ssh connections can disconnect due to network or hardware problems. Any processes initialized through that connection will be "hung up" and terminate. Running a command with nohup insures that the command will not be hung up if the shell is closed or if the network connection fails.

Run yes (continually outputs "y" until it's killed) with nohup:



```
[ andrew@pc01 ~ ]$ nohup yes &
[1] 13173
ps shows a list of the current user's processes (note PID number 13173):
[ andrew@pc01 ~ ]$ ps | sed -n '/yes/p'
13173 pts/10    00:00:12 yes
...log out and log back into this shell...
```

The process has disappeared from ps!

```
[ andrew@pc01 ~ ]$ ps | sed -n '/yes/p'
```

But it still appears in top and htop output:

```
[ andrew@pc01 ~ ]$ top -bn 1 | sed -n '/yes/p'
13173 andrew    20   0   4372    704    636 D   25.0   0.0   0:35.99 yes
Kill this process with -9 followed by its process ID (PID) number:
[ andrew@pc01 ~ ]$ kill -9 13173
It no longer appears in top, because it's been killed:
[ andrew@pc01 ~ ]$ top -bn 1 | sed -n '/yes/p'
```

cron / crontab / >>  
[ Back to Table of Contents ]

cron provides an easy way of automating regular, scheduled tasks.

You can edit your cron jobs with crontab -e (opens a text editor). Append the line:

```
* * * * * date >> ~/datefile.txt
This will run the date command every minute, appending (with the >> operator) the output to a file:
[ andrew@pc02 ~ ]$ head ~/datefile.txt
Sat Jan 12 14:37:01 GMT 2019
Sat Jan 12 14:38:01 GMT 2019
Sat Jan 12 14:39:01 GMT 2019...
```

Just remove that line from the crontab file to stop the job from running. cron jobs can be set up to run at particular minutes of each hour (0-59), particular hours of each day (0-23), particular days of each month (1-31), particular months of each year (1-12), or particular days of each week (0-6, Sun-Sat). This is what the five stars at the beginning of the command above represent, respectively. Replace them with specific numbers to run them on particular days or at particular times.

If a job is to be run irrespective of, for instance, the day of the week, then the position that represents the day of the week (the 5th position) should contain a star (\*). This is why the command above runs every minute (the smallest interval available). cron jobs can be set up to run only when the system is rebooted, with @reboot replacing the stars/numbers. Jobs can also be run a specific number of times per hour or day or at multiple specific times per hour / day / week / month / etc.

Check out this tutorial for more info.

## Miscellaneous

pushd / popd  
[ Back to Table of Contents ]

Use pushd and popd to maintain a directory stack, instead of cd-ing everywhere.

Start in the home directory -- this will be the bottom directory in our "stack":

```
[ andrew@pc01 ~ ]$ pwd
/home/andrew
Move to this directory with a long name, "push" it onto the stack with pushd:
[ andrew@pc01 ~ ]$ pushd /etc/java/security/security.d/
/etc/java/security/security.d ~
Move to a third directory and add it to the stack:
```

```
[ andrew@pc01 security.d ]$ pushd ~/test/
~/test /etc/java/security/security.d ~
When a new directory is added to the stack, it is added to the left-hand side of the list printed by pushd. To "pop" the top directory off (return to the most recent directory we added), we can use the popd command.
```

"Pop" off the top directory, move to the next one down the stack with popd:

```
[ andrew@pc01 test ]$ popd
/etc/java/security/security.d ~
```





nautilus / date / cal / bc  
[ Back to Table of Contents ]

nautilus initialises a GUI remote desktop session and opens a file browser.

date shows the current date and time:

```
[ andrew@pc01 ~ ]$ date  
Fri Jan 11 17:40:30 GMT 2019
```

cal shows an ASCII calendar of this month with today's date highlighted:

```
[ andrew@pc01 ~ ]$ cal  
January 2019
```

```
Su Mo Tu We Th Fr Sa  
      1  2  3  4  5  
 6  7  8  9 10 11 12  
13 14 15 16 17 18 19  
20 21 22 23 24 25 26  
27 28 29 30 31
```

bc is a basic arithmetic calculator (use Python instead):

```
[ andrew@pc01 ~ ]$ bc  
bc 1.06.95 ...  
20/4  
5
```